

भारतीय  
सूचना प्रौद्योगिकी  
संस्थान वडोदरा



Indian  
Institute of Information  
Technology Vadodara

(शिक्षा मंत्रालय, भारत सरकार के अधीन राष्ट्रीय महत्व का संस्थान)  
(An Institution of National Importance under Ministry of Education, Govt. of India)

# B.Tech. Curriculum - 2026



<b>S.No.</b>	<b>Content</b>	<b>Page</b>
1.	B.Tech. Programme: Regulations	1-4
2.	B.Tech. Programme: Credit Distribution	5
3.	Courses: First Year (Semester I & II)	6-7
4.	Courses: Semester III to VIII	
	B.Tech. in ACP	8-11
	B.Tech. in AI	12-13
	B.Tech. in CSE	14-15
	B.Tech. in CSE (Cyber Security)	16-17
	B.Tech. in ECE	18-21
	B.Tech. in IT	22-23
	B.Tech. in MAC	24-26
5.	Courses and Contents	
	Semester-I	27-38
	Semester-II	39-51
	Semester-III	52-64
	Semester-IV	65-83
	Semester-V	84-111
	Semester-VI	112-113
	Semester-VII	114
	Semester-VIII	115-116
6.	Appendix-A	117-132

## B.Tech. Programme

The Institute offers 4-years B.Tech. programme in the following branches:

1. Applied and Computational Physics (ACP)
2. Artificial Intelligence (AI)
3. Computer Science and Engineering (CSE)
4. Computer Science and Engineering (Cybersecurity) (CCS)
5. Electronics and Communication Engineering (ECE)
6. Information Technology (IT)
7. Mathematics and Computing (MAC)

### Academic Session

Each academic session is divided into two semesters of approximately eighteen weeks duration (with at-least seventy working days for classes in each semester):

1. Autumn Semester (July-November)
2. Winter Semester (January-April)

In addition, there may be a semester during the summer break, called a Summer Semester (May-June). The Senate approves schedule of academic activities for a session, inclusive of dates for registration, mid-semester and end-semester examinations, inter-semester breaks etc. The schedule is laid down in the Academic Calendar for the session.

### Registration

At the beginning of each semester, until the completion of the program, a student must register for the semester and for the courses that he/she will study during the semester.

### Eligibility for Course Registration

A student with no backlog courses (*i.e.* who has passed all the previous courses) will be eligible to register for all courses prescribed in the curriculum for semester. A student who has backlog course(s) or is on academic probation may be recommended a different set of courses by the Dean of Academic Programs (Dean-AP).

### Prerequisite Courses

A student registering for a course must have successfully completed the prerequisite course(s), if any, for that particular course. For hard prerequisite, a minimum grade of 'DD' is required.

### Auditing of Courses

Auditing of courses allows students to gain exposure to additional subjects without increasing their overall workload. Registration of courses for Audit is permitted from fifth semester onwards under the following conditions:

1. A student can audit a maximum of two courses during the entire program.

2. A student has to enter the courses to be audited in the Course Registration Form while registering for the semester. The word 'Audit' would be specially mentioned in the remarks column of the student's course registration form.
3. A student can register a course for audit provided the following two conditions are satisfied: (i) the course instructor permits and approves the registration, and (ii) the lecture, lab and tutorial time-table strictly permit.
4. An audit course will not be considered as an overload.
5. If the student's performance is satisfactory, a grade of 'P' (Pass) would be awarded. If the performance is not satisfactory, 'F' (Fail) would be awarded.
6. An audit course will not be considered for the calculation of Semester Performance Index (SPI) / Cumulative Performance Index (CPI). However, the course will be reflected in the Semester Grade Report and Transcript as an Audit Course provided a grade of 'P' was obtained, otherwise the course will not appear in the Semester Grade Report and Transcript.

### Definition of Credits

Teaching of the courses shall be reckoned in credits; credits are assigned to the courses based on the following general pattern:

1 hour of Lecture (L) per week	1 Credit
1 hour of Tutorial (T) per week	1 Credit
2 hours of Laboratory (P) per week	1 Credit
3 hours of Laboratory (P) per week	2 Credits
3 hours of Project/Research /Internship	1 Credit

### Course Load in Regular Semesters

A student is permitted to register for additional courses over the prescribed courses in the curriculum for a regular semester provided the total number of courses does not exceed 7 and the total credits do not exceed 26. A student is permitted to under-load his/her prescribed academic load in a regular semester by dropping one or more courses provided the number of courses is at least 4 and the registered credits are not less than 12. However, after completion of his/her seventh regular semester, a student will be permitted to register for less than four courses.

### Course Assessment

The assessment of students' academic performance include in-semester and end-semester examinations along with other continuous evaluation components. The various components of continuous assessment in a course may include home assignments, tutorial assignments, group assignments, quizzes, tests (open or

closed book), viva-voce, mini projects, etc. Attendance in lectures/ tutorials/ labs may also be given due weightage in course assessment. The instructor may make attendance in lectures/ tutorials/ labs compulsory (80% or less). The instructor may, in due consultation with the Dean-AP, award 'F' grade to students who do not achieve the prescribed level of attendance in that course.

The distribution of weightage, for the assessment of academic performance of students in a course, through various modes listed above will be communicated by the course instructor at the beginning of the semester with due approval from the Director.

[*Note: Academic requirements such as projects and summer assignments, which are prescribed in the curriculum, are regarded as courses for the purpose of assessment.*]

### Letter Grade (10-point Scale)

For every course registered by a student, he/she is awarded a letter grade based on his/her combined performance in all the assessments. These letter grades are assigned points on a 10-point scale as described in the table below:

Letter Grade	Points	Remark
AA	10	Outstanding
AB	9	Excellent
BB	8	Very Good
BC	7	Good
CC	6	Average
CD	5	Below Average
DD	4	Poor
F	0	Fail
P	-	Pass
I	-	Incomplete

A student passes the course if he/she gets any grade in the range of 'AA' to 'DD', but fails if he/she gets the grade 'F'. Certain courses are indicated as Pass/Fail courses, and in these courses a grade of 'P' or 'F' is awarded. 'F' grade may also be awarded in case of malpractice in examination/continuous evaluation process. Pass/Fail courses are not considered for calculation of SPI/CPI.

'I' grade will be awarded in a course if the overall performance of the student is satisfactory in the course, but the student either misses the end-semester examination due to illness, accident/death in the family or obtains such an approval from the Dean-AP under

exceptional circumstances. A student who misses the end-semester examination must apply for permission with reasoning and proof. An application not so supported will not be considered. Grade 'I' awarded for missing the end-semester examination will be converted into a performance grade (depending on the overall performance of the student in the course) after taking an examination equivalent to the end-semester examination of that particular course. An 'I' grade must be converted into a performance grade by the specified date in the academic calendar for the next semester, otherwise it will be converted into 'F' grade.

### Performance Indices

#### [A] Semester Performance Index (SPI)

The performance of a student in a semester is indicated by the *SPI*. The *SPI* is the weighted average of the grade points obtained in all the courses registered by the student during the semester, calculated to two decimal places.

#### [B] Cumulative Performance Index (CPI)

An up-to-date assessment of the overall performance of a student from the time of entering the Institute is obtained by calculating the student's *CPI*. The *CPI* is weighted average of the grade points obtained in all the courses registered for credit by the student after entering the Institute. The *CPI* is also calculated to two decimal places.

#### Calculation of SPI and CPI

Let the course credits are  $U_1, U_2, \dots$  and the numeric values of the corresponding grade awarded in the courses are  $G_1, G_2, \dots$ , respectively, the SPI is given by

$$SPI = \frac{U_1 G_1 + U_2 G_2 + \dots}{U_1 + U_2 + \dots}$$

In the above computation, the courses with 'P' grade are not considered. Similarly, the *CPI* indicates the cumulative academic performance in all the courses taken including those taken in the current semester as

$$CPI = \frac{1}{Total\ Credits} \sum_{i=1}^8 (SPI \times Total\ credits\ of\ i^{th}\ semester)$$

#### Graduating CPI and Class

For the purposes of computing the *CPI* at the end of the B. Tech. program, the student's *CPI* will be computed on the basis of the best *CPI* obtained from the courses taken. The grade of B. Tech. Project (if graded as prescribed in the curriculum) will be included while computing the final *CPI* of the student. The minimum *CPI* for eligible to graduate the B. Tech. program is 5.00. The Transcript of a graduating student will indicate

1. *Distinction* when  $CPI \geq 9.00$ ,
2. *First Class* when  $6.50 \leq CPI < 9.00$  and
3. *Pass* when  $5.00 \leq CPI < 6.50$ .

### Repeating a Course

#### [A] As a Backlog Course

A student must repeat a course taken for credit in which he/she has obtained 'F' grade. Such a course is regarded as a backlog course. A backlog elective course can be replaced by another elective of the same category. A student can appear examinations (In-Semester and End-Semester) components only as a backlog course. The component of continuous evaluation will be carried forward from earlier evaluation.

#### [B] For Grade Improvement

A student whose *CPI* is less than 5.00 can be permitted to reappear in the courses in which a 'DD' grade was obtained. This is for the purpose of grade improvement in a regular semester only. The grade obtained in the repeated attempt(s) will be considered for the purpose of calculating the *CPI* for the semesters thereafter. However, the grade obtained in the first and subsequent attempt(s) will be shown in the Transcript.

#### Award of Degree

The B. Tech. (CS) and B. Tech. (IT) degree will be conferred on a student after he/she has fulfilled the graduation requirements stipulated in the curriculum (as approved by the senate).

#### Award of Medals

**[A] The Chairperson's Gold Medal** Graduate(s) excelling in academic performance with  $CPI (>7.00)$  at the end of 8th semester without any backlog and overall contribution to the Institute Corporate activities comprising Campus Life, Social Services and Evidenced Leadership qualities would be considered for the award of Chairperson's Gold Medal. In addition to the above criteria, the student must not have been awarded any deterrent by the Institute Disciplinary Action Committee throughout the program. Based on an overall assessment, following due process of evaluation, recommendation of the Senate and approval of the Board of Governors (BoG) of the Institute, a candidate would be considered for the award of Chairperson's Gold Medal, in accordance with the Institute Rules.

The evaluation for the award will be based on the following key aspects:

	Indicator	Details	Maximum
1.	Academic Performance	Academic excellence. <i>CPI</i> (7-8: 10 point.	25

		8-9: 15 points: 9-10: 20 points, highest in batch 25 points)	
2.	Sports & Extra-Curricular Activities	Participation and achievements in inter-college and national/international events. (participation per event 2. per awards/achievements 5)	25
3.	Technical Research and Innovation	Publications/ Patents/Startup/ Other technical projects of impact (5 per contribution). Hackathon (2 per participation and 5 for being in top three per event). Other technical Events (2 per participation and 3 per achievement/award/ recognition).	25
4.	Leadership. Outreach & Social Work	Holding positions of responsibility in student organizations. placement cell. clubs, or technical societies.. Contributions to community service. NGOs. and social impact initiatives. (holding position of responsibility/ participation 5 points per position/ participation. successful conduct of an event 5 points per 25 event)	25

**Note:**

1. Total points accumulated for each indicator will be limited by the respective award category threshold.
2. The above guideline is indicative. However, the scrutiny committee may propose criteria for selection with modifications/ changes.

**[B] The Institute Medals**

1. The Institute *Gold Medals* will be awarded to students who have secured first places in their respective programs.
2. The Institute *Silver Medals* will be awarded to the second place holders for each program.

**Poor Academic Performance****[A] Academic Probation**

A student will be placed on Academic Probation during his/her second semester with written intimation if his/her *SPI* at the end of first semester is less than 4.50. In subsequent semesters, a student will be placed on Academic Probation with written intimation if his/her *CPI* in the previous semester is less than 5.00 or if his/her *SPI* is less than 4.50 in the previous semester. For every student placed on Academic Probation, the Dean-AP will prescribe a minimum *SPI* the student must attain in the semester. The minimum *SPI* so stipulated will be arrived at on the basis of the performance of the student in terms of her/his *SPI/CPI* as compared to the minimum requirements for graduation.

**[B] Discontinued from the Institute on Account of Poor Academic Performance**

If the performance of a student is consistently poor and that he/she is not likely to benefit from continuing in the program, he/she would be required to leave the Institute. For this purpose an assessment of the student's academic performance will initially be made at the end of the second semester of his/her stay at the Institute and thereafter at the end of every subsequent semester. This assessment will be based on the *CPI* and *SPI* obtained by the student.

**[C] Discontinued from the Institute on account of Poor Academic Performance at the end of the Second and Fourth Semester**

A student whose *CPI* is less than 4.00 at the end of second or fourth semester shall be discontinued from the Institute. However, such a student may be allowed to register for the available backlog courses if offered in the summer semester following the second or fourth semester. Such a student is permitted to register for a maximum of three of the available summer courses in which he/she is having 'F' or 'DD' grade. In case the student achieves the minimum *CPI* of 4.00 at the end of the relevant summer semester, he/she will be allowed to continue the program.

**Minimum and Maximum Period for Completion of B. Tech. Program**

The minimum period to complete the program is four academic years. In any case, a student should fulfil the requirements for her/his degree within a maximum period of six academic years, failing which she/he will be required to leave the Institute. The period of six years excludes any semester in which the student has availed "withdrawn" status.

**Certificate of Academic Accomplishment**

A student who is unable to complete the degree requirements within the stipulated maximum period would be eligible to receive a "Certificate of Academic Accomplishment" by applying for it. The eligibility criteria and procedure for issue of the Certificate would be as laid down by the Institute senate from time to time.

---

## B.Tech. Programme: Credit Distribution

S.No.	Area	Credits Approved	Branch of Studies						
			CSE	IT	ECE	AI	MAC	ACP	CCS
1	Mathematics & Statistics + Natural Science (Physics, Chemistry, Biology), Environmental Science (12+12)	24 ± 3	26	24	24	27	24	25	21
2	Humanities, Social Science, Literature, Management & Soft-Skills + Open Electives + IKS + NCC/ NSS/ NCA/ NSO + Skill Enhancement (Disaster Management) (12+6+2+0+2)	22 ± 3	22	22	22	23	25	24	23
3	Core Engineering Courses + Program Elective (from the branch of study) (60+20)	80 ± 4	77	77	84	78	78	78	79
4	Core Engineering Courses + Elective (from the other branch of Engineering) (12+6)	18 ± 4	22	22	17	17	18	19	18
5	Projects, Internships (Research/Industrial)	18	18	18	18	18	18	18	18
<b>6</b>	<b>Total Credits</b>	<b>162 ± 3</b>	<b>165</b>	<b>163</b>	<b>165</b>	<b>163</b>	<b>163</b>	<b>164</b>	<b>159</b>
7	Honors in *: Additional courses in specific verticals (like AI, Data Science, Software Engineering, Security, etc.)	12	12	12	12	12	12	12	12
8	Minor in *: Mathematics, Physics, etc.	12	12	12	12	12	12	12	12
<b>9</b>	<b>Total Credits (with Hons)</b>	<b>174 ± 3</b>	<b>177</b>	<b>175</b>	<b>175</b>	<b>175</b>	<b>175</b>	<b>176</b>	<b>171</b>
<b>10</b>	<b>Total Credits (with Minor)</b>	<b>174 ± 3</b>	<b>177</b>	<b>175</b>	<b>177</b>	<b>175</b>	<b>175</b>	<b>176</b>	<b>171</b>

**CSE:** Computer Science and Engineering; **IT:** Information Technology; **ECE:** Electronics and Communication Engineering; **AI:** Artificial Intelligence; **MAC:** Mathematics and Computing; **ACP:** Applied and Computational Physics; **CCS:** Computer Science and Engineering (Cyber Security).

## Courses: First Year

SEMESTER - I				Branch of Studies						
S.No.	Course Code	Course Name	L-T-P: C	CSE	IT	ECE	AI	MAC	ACP	CCS
1	MA101	Mathematics-I	3-1-0: 4							
2	MC101	Combinatorial Linear Algebra	1-0-0: 1							
3	PH101	Physics-I	3-1-0: 4							
4	PH161	Experimental Physics-I	0-0-2: 1							
5	CS101	Computer Programming and Problem Solving	3-0-0: 3							
6	CS161	Computer Programming and Problem Solving Lab	0-0-3: 2							
7	EC101	Basic Electronics-I	3-1-0: 4							
8	EC161	Basic Electronics Lab-I	0-0-3: 2							
9	EC103	Introduction to ECE	1-0-0: 1							
10	CC101	Cybersecurity Essentials	3-0-0: 3							
11	CC103	System Administration and Maintenance	0-1-2: 2							
12	CC105	Digital Design and Computer Architecture	3-0-2: 4							
13	AI101	Introduction to Artificial Intelligence	2-0-0: 2							
14	HS101	Spoken and Written Communication	2-0-2: 3							
15	HS103	Artificial Intelligence and Ethics	2-0-0: 2							
16	HS105	History of Mathematics and Computing	2-0-0: 2							
17	IK101	Introduction to Yoga	0-2-0: 2							
18	NA101	NCC/NSS/NCA/NSO-I	0-0-0: 0							
<b>Total Credits</b>				<b>25</b>	<b>25</b>	<b>24</b>	<b>24</b>	<b>22</b>	<b>25</b>	<b>23</b>

**Note:** For a branch of study, the courses indicated by the filled blocks (in green color) are prescribed in Semester-I.

SEMESTER - II				Branch of Study						
S.No.	Course Code	Course Name	L-T-P: C	CSE	IT	ECE	AI	MAC	ACP	CCS
1	MA102	Mathematics-II (Discrete Maths)	3-1-0: 4							
2	MA104	Mathematics-II (Calculus)	3-1-0: 4							
3	MA106	Optimization Techniques	3-0-0: 3							
4	MC102	Calculus	3-1-0: 4							
5	PH102	Physics-II	3-1-0: 4							
6	PH162	Experimental Physics-II	0-0-2: 1							
7	PH104	Physics-III	3-0-0: 3							
8	CS102	Introduction to Data Structures	3-0-0: 3							
9	CS162	Introduction to Data Structures Lab	0-1-2: 2							
10	AI104	Data Visualization and Analytics	0-1-2: 2							
11	EC102	Basic Electronics-II	3-1-0: 4							
12	EC162	Basic Electronics Lab-II	0-0-3: 2							
13	CC102	Python Programming	1-0-3: 3							
14	CC104	Modern Cryptography	3-0-2: 4							
15	HS102	Science Technology and Society	3-0-0: 3							
16	HS104	Disaster Management	2-0-0: 2							
17	NA102	NCC/NSS/NCA/NSO-I	0-0-0: 0							
<b>Total Credits</b>				<b>25</b>	<b>25</b>	<b>25</b>	<b>25</b>	<b>23</b>	<b>25</b>	<b>21</b>

**Note:** For a branch of study, the courses indicated by the filled blocks (in green color) are prescribed in Semester-II.

## Courses: Semester III to VIII

### 1. B.Tech. in Applied and Computational Physics

Semester	Course Code	Domain	Course Name	L	T	P	C
Semester - III	MA201	MA	Mathematics-III	3	1	0	4
	PH201	PH	Optics	2	1	2	4
	HS201	SE	Technical Writing	1	1	2	3
	CS201	CS	Object-Oriented Design & Programming	3	0	0	3
	CS261	CS	Object-Oriented Design & Programming Lab	0	0	2	1
	CS203	CS	Design and Analysis of Algorithms	3	0	0	3
	CS263	CS	Design and Analysis of Algorithms Lab	0	0	2	1
	EC201	EC	Digital Logic Design	3	0	0	3
	EC261	EC	Digital Logic Design Lab	0	0	2	1
				<b>15</b>	<b>3</b>	<b>10</b>	<b>23</b>
Semester - IV	MA202	MA	Numerical Techniques	0	1	2	2
	HS202	HS	Economics	3	0	0	3
	PH202	PH	Foundations of Quantum Technologies	3	0	0	3
	CS204	CS	Database Management System	3	0	0	3
	CS264	CS	Database Management System Lab	0	0	2	1
	PH262	PH	Laboratory for Quantum Technologies	0	0	2	1
	CS208	CS	Computer Organization and Architecture	3	0	0	3
	CS268	CS	Computer Organization and Architecture Lab	0	0	2	1
				<b>12</b>	<b>1</b>	<b>8</b>	<b>17</b>
Semester - V	CS301	CS	Computer Networks	3	0	0	3
	CS361	CS	Computer Networks Lab	0	0	2	1
	CS303	CS	Operating Systems	3	0	0	3
	CS363	CS	Operating Systems Lab	0	0	2	1
	EC307	EC	Introduction to VLSI Design	3	0	2	4
	PH301	PH	Computational Physics	3	0	0	3
	PH361	PH	Computational Physics Lab	0	0	2	1
	PH303	PH	Thermal and Statistical Physics	3	0	0	3
	HS301	HS	Entrepreneurship and IP Strategy	0	1	0	1
	DP391	PI	Design Project	0	0	0	3

				<b>15</b>	<b>1</b>	<b>8</b>	<b>23</b>
Semester - VI	-	PH	Program Elective (PE1)*	3	0	2	4
	-	PH	Program Elective (PE2)*	3	0	0	3
	-	PH	Program Elective (PE3)*	3	0	2	4
	-	PH	Program Elective (PE4)*	0	1	2	2
	-	EO	Elective from other Branch (EO1)	3	0	2	4
	-	OE	Open Elective (OE1)	3	0	0	3
				<b>15</b>	<b>1</b>	<b>8</b>	<b>20</b>
Semester - VII	PH401	PH	Physics of Atoms and Molecules	3	0	0	3
	-	PH	Program Elective (PE5)*	3	0	0	3
	-	EO	Elective from other Branch (EO2)	3	0	2	4
	-	OE	Open Elective (OE2)	3	0	0	3
	SI491	PI	Research/Industrial Internship	0	0	0	3
				<b>12</b>	<b>0</b>	<b>2</b>	<b>16</b>
Semester - VIII	BP490	PI	B.Tech. Project	0	0	36	12
				<b>0</b>	<b>0</b>	<b>36</b>	<b>12</b>

### \*Tentative List of Programme Electives for B.Tech. in ACP (PE1 - PE5)

Programme Verticals	Courses
<b>Quantum Technologies</b>	Condensed Matter Physics
	Computational Condensed Matter Physics
	Machine Learning in Materials Science and Atomistic Modelling
	Introduction to Quantum Computation
	Introduction to Quantum Algorithms
	Introduction to Open Quantum Systems
	Introduction to Lie Groups and Representations
	Advanced Quantum Mechanics
	Complex analysis and Approximation Methods
<b>Solid-State Materials and Devices</b>	Physics of Semiconductor Devices
	Electronic Properties of Materials
	Science and Technology of Thin Films

	Nanostructured Materials and Devices
	Energy Materials and Devices
	Introduction to Spintronics
	Characterisation Techniques for Materials
	Organic Semiconductors and Devices
	New and Emerging Semiconductors
	Flexible Electronics
<b>Optics &amp; Photonics</b>	Nonlinear Optics
	Introduction to Quantum Optics
	Fundamentals of Optoelectronics
	Laser & Plasma Technology
	Introduction of Quantum Information Science
	Laser systems and applications
	Advanced Fourier Optics and Holography
	Optical System Design
	Computational Optics
	Fibre Optics & Optical Communication
<b>Energy &amp; Environment</b>	Introduction to Energy and Environment
	Introduction to Non-Conventional Sources of Energy
	Optimal Design of Energy Systems
	Plasmas for Energy and Environment
	Nanomaterials for Energy Applications
	Hydrogen Energy and Fuel Cell Technology
	Battery Technology for EV
	Distributed and Decentralised Energy Systems
	Special Topics on Emerging Trends of Energy & Environmental Technologies

	Photovoltaic Technologies
<b>Experimental &amp; Computational Physics</b>	Nonlinear Dynamics
	Vacuum Science and Technology
	Computational Fluid Dynamics
	Modelling and Simulation of Physical Systems
	Computational Plasma Physics
	Multiscale and Multiphysics Simulation
	Computational Geophysics

## 2. B.Tech. in Artificial Intelligence

Semester	Course Code	Domain	Course Name	L	T	P	C
Semester - III	MA201	MA	Mathematics-III	3	1	0	4
	AI201	CS	Operating Systems	3	0	0	3
	AI261	CS	Operating Systems Lab	0	0	2	1
	HS201	SE	Technical Writing	1	1	2	3
	CS203	CS	Design and Analysis of Algorithms	3	0	0	3
	CS263	CS	Design and Analysis of Algorithms Lab	0	0	2	1
	AI203	CS	Linear Systems and Signal Processing	2	0	2	3
	SC201	EC	Environmental Science	2	0	0	2
				<b>14</b>	<b>2</b>	<b>8</b>	<b>20</b>
Semester - IV	AI202	CS	Machine Learning	3	0	2	4
	HS202	HS	Economics	3	0	0	3
	CS204	CS	Database Management System	3	0	0	3
	CS264	CS	Database Management System Lab	0	0	2	1
	AI206	CS	Computer Organization and Architecture	3	0	0	3
	AI266	CS	Computer Organization and Architecture Lab	0	0	2	1
	AI204	CS	Artificial Intelligence	3	0	0	3
	AI264	CS	Artificial Intelligence Lab	0	0	2	1
				<b>15</b>	<b>0</b>	<b>8</b>	<b>19</b>
Semester - V	AI301	CS	Control Systems	3	1	0	4
	AI303	CS	Deep Learning	3	0	2	4
	AI305	CS	Natural Language Processing	3	0	2	4
	AI307	CS	Computer Vision	3	0	2	4
	CS309	CS	Introduction to Cryptography and Network Security	3	0	0	3
	CS369	CS	Introduction to Cryptography and Network Security Lab	0	0	2	1
	HS301	HS	Entrepreneurship and IP Strategy	0	1	0	1
	DP391	PI	Design Project	0	0	0	3
				<b>15</b>	<b>2</b>	<b>8</b>	<b>24</b>
Semester - VI	-		Program Elective (PE1)*	3	0	2	4
	-		Program Elective (PE2)*	3	0	0	3
	-		Program Elective (PE3)*	3	0	2	4

	-		Program Elective (PE4)*	0	1	2	2
	-	EO	Elective from other Branch (EO1)	3	0	0	3
	-	OE	Open Elective (OE1)	3	0	0	3
				<b>15</b>	<b>1</b>	<b>6</b>	<b>19</b>
Semester - VII	-		Program Elective (PE5)*	3	0	0	3
	-		Program Elective (PE6)*	0	1	3	3
	-		Program Elective (PE7)*	3	0	0	3
	-		Program Elective (PE8)*	0	0	2	1
		EO	Elective from other Branch (EO2)	3	0	0	3
	-	OE	Open Elective (OE2)	3	0	0	3
	SI491	PI	Research/Industrial Internship	0	0	0	3
				<b>12</b>	<b>1</b>	<b>5</b>	<b>19</b>
Semester - VIII	BP490	PI	B.Tech. Project	0	0	36	12
				<b>0</b>	<b>0</b>	<b>36</b>	<b>12</b>

**\*Programme Electives for B.Tech. in AI (PE1 - PE8) is provided in Appendix A.**

### 3. B.Tech. in Computer Science and Engineering

Semester	Course Code	Domain	Course Name	L	T	P	C
Semester - III	MA201	MA	Mathematics-III	3	1	0	4
	HS201	SE	Technical Writing	1	1	2	3
	CS201	CS	Object-Oriented Design & Programming	3	0	0	3
	CS261	CS	Object-Oriented Design & Programming Lab	0	0	2	1
	CS203	CS	Design and Analysis of Algorithms	3	0	0	3
	CS263	CS	Design and Analysis of Algorithms Lab	0	0	2	1
	EC201	EC	Digital Logic Design	3	0	0	3
	EC261	EC	Digital Logic Design Lab	0	0	2	1
	SC201	HS	Environmental Science	2	0	0	2
				<b>15</b>	<b>2</b>	<b>8</b>	<b>21</b>
Semester - IV	MA202	MA	Numerical Techniques	0	1	2	2
	HS202	HS	Economics	3	0	0	3
	CS202	CS	System Software	3	0	0	3
	CS204	CS	Database Management System	3	0	0	3
	CS264	CS	Database Management System Lab	0	0	2	1
	CS206	CS	Software Engineering	2	0	0	2
	CS266	CS	Software Engineering Lab	0	1	2	2
	CS208	CS	Computer Organization and Architecture	3	0	0	3
	CS268	CS	Computer Organization and Architecture Lab	0	0	2	1
				<b>14</b>	<b>2</b>	<b>8</b>	<b>20</b>
Semester - V	CS301	CS	Computer Networks	3	0	0	3
	CS361	CS	Computer Networks Lab	0	0	2	1
	CS303	CS	Operating Systems	3	0	0	3
	CS363	CS	Operating Systems Lab	0	0	2	1
	CS305	CS	Formal Language and Automata Theory	3	1	0	4
	CS307	CS	Artificial Intelligence	3	0	0	3
	CS367	CS	Artificial Intelligence Lab	0	0	2	1
	CS309	CS	Introduction to Cryptography and Network Security	3	0	0	3
	CS369	CS	Introduction to Cryptography and Network Security Lab	0	0	2	1
	HS301	HS	Entrepreneurship and IP Strategy	0	1	0	1

	DP391	PI	Design Project	0	0	0	3
				<b>15</b>	<b>2</b>	<b>8</b>	<b>24</b>
Semester - VI	CS401	CS	Distributed and Parallel Computing	3	0	0	3
	CS461	CS	Distributed and Parallel Computing Lab	0	0	2	1
	-	CS	Program Elective (PE1)*	3	0	2	4
	-	CS	Program Elective (PE2)*	3	0	0	3
	-	CS	Program Elective (PE3)*	0	1	2	2
	-	EO	Elective from other Branch (EO1)	3	0	0	3
	-	OE	Open Elective (OE1)	3	0	0	3
				<b>15</b>	<b>1</b>	<b>6</b>	<b>19</b>
Semester - VII	-	CS	Program Elective (PE4)*	3	0	2	4
	-	CS	Program Elective (PE5)*	3	0	0	3
	-	CS	Program Elective (PE6)*	3	0	2	4
	-	EO	Elective from other Branch (EO2)	3	0	0	3
	-	OE	Open Elective (OE2)	3	0	0	3
	SI491	PI	Research/Industrial Internship	0	0	0	3
				<b>15</b>	<b>0</b>	<b>4</b>	<b>20</b>
Semester - VIII	BP490	PI	B.Tech. Project	0	0	36	12
				<b>0</b>	<b>0</b>	<b>36</b>	<b>12</b>

**\*Programme Electives for B.Tech. in CSE (PE1 - PE6) is provided in Appendix A.**

#### 4. B.Tech. in Computer Science and Engineering (Cyber Security)

Semester	Course Code	Domain	Course Name	L	T	P	C
Semester - III	MA201	MA	Mathematics-III	3	1	0	4
	HS201	SE	Technical Writing	1	1	2	3
	CS201	CS	Object-Oriented Design & Programming	3	0	0	3
	CS261	CS	Object-Oriented Design & Programming Lab	0	0	2	1
	CS203	CS	Design and Analysis of Algorithms	3	0	0	3
	CS263	CS	Design and Analysis of Algorithms Lab	0	0	2	1
	CC201	CS	Networking Fundamentals	3	0	2	4
	SC201	HS	Environmental Science	2	0	0	2
				<b>15</b>	<b>2</b>	<b>8</b>	<b>21</b>
Semester - IV	HS202	HS	Economics	3	0	0	3
	CC202	CS	Software and Hardware Security	3	0	2	4
	CS204	CS	Database Management System	3	0	0	3
	CS264	CS	Database Management System Lab	0	0	2	1
	CS206	CS	Software Engineering	2	0	0	2
	CS264	CS	Software Engineering Lab	0	1	2	2
	CS208	CS	Computer Organization and Architecture	3	0	0	3
	CS268	CS	Computer Organization and Architecture Lab	0	0	2	1
				<b>14</b>	<b>1</b>	<b>8</b>	<b>19</b>
Semester - V	CC301	CS	Web Technologies	3	0	2	4
	CC303	CS	Operating Systems and Linux for Cybersecurity	3	0	2	4
	CC305	CS	Cybersecurity Threats and Defense Mechanisms	3	0	2	4
	CC307	CS	Fundamentals of Machine Learning	3	0	2	4
	CC309	CS	Embedded Systems and Internet of Things	3	0	2	4
	HS301	HS	Entrepreneurship and IP Strategy	0	1	0	1
	DP391	PI	Design Project	0	0	0	3
				<b>15</b>	<b>1</b>	<b>10</b>	<b>24</b>
Semester - VI	-	CS	Program Elective (PE1)*	3	0	2	4
	-	CS	Program Elective (PE2)*	3	0	0	3
	-	CS	Program Elective (PE3)*	3	0	2	4
	-	CS	Program Elective (PE4)*	0	1	2	2

	-	EO	Elective from other Branch (EO1)	3	0	0	3
	-	OE	Open Elective (OE1)	3	0	0	3
				<b>15</b>	<b>1</b>	<b>6</b>	<b>19</b>
Semester - VII	-	CS	Program Elective (PE5)*	3	0	0	3
	-	CS	Program Elective (PE6)*	3	0	2	4
	-	CS	Program Elective (PE7)*	3	0	2	4
	-	EO	Elective from other Branch (EO2)	3	0	0	3
	-	OE	Open Elective (OE2)	3	0	0	3
	SI491	PI	Research/Industrial Internship	0	0	0	3
				<b>15</b>	<b>0</b>	<b>4</b>	<b>20</b>
Semester - VIII	BP490	PI	B.Tech. Project	0	0	36	12
				<b>0</b>	<b>0</b>	<b>36</b>	<b>12</b>

**\*Programme Electives for B.Tech. in CCS (PE1 - PE7) is provided in Appendix A.**

## 5. B.Tech. in Electronics and Communication Engineering

Semester	Course Code	Domain	Course Name	L	T	P	C
Semester - III	MA201	MA	Mathematics-III	3	1	0	4
	HS201	SE	Technical Writing	1	1	2	3
	EC201	EC	Digital Logic Design	3	0	0	3
	EC261	EC	Digital Logic Design Lab	0	0	2	1
	EC203	EC	Linear Systems and Signal Processing	1	1	2	3
	EC205	EC	Antenna and Wave Propagation	3	0	0	3
	EC265	EC	Antenna and Wave Propagation Lab	0	0	2	1
	EC207	EC	Network Analysis and Synthesis	2	0	0	2
	SC201	HS	Environmental Science	2	0	0	2
				<b>15</b>	<b>3</b>	<b>8</b>	<b>22</b>
Semester - IV	HS202	HS	Economics	3	0	0	3
	EC202	EC	IC Design	3	1	0	4
	EC262	EC	IC Design Lab	0	0	2	1
	EC204	EC	Principles of Communication	3	0	0	3
	EC264	EC	Principles of Communication Lab	0	1	2	2
	EC206	EC	Control Systems	3	1	0	4
	CS208	CS	Computer Organization and Architecture	3	0	0	3
	CS268	CS	Computer Organization and Architecture Lab	0	0	2	1
				<b>15</b>	<b>3</b>	<b>6</b>	<b>21</b>
Semester - V	EC301	EC	Computer Networks	3	0	0	3
	EC361	EC	Computer Networks Lab	0	0	2	1
	EC303	EC	Digital Signal and Image Processing	3	0	0	3
	EC363	EC	Digital Signal and Image Processing Lab	0	0	2	1
	EC305	EC	Digital Communication	3	0	0	3
	EC365	EC	Digital Communication Lab	0	0	2	1
	EC307	EC	Introduction to VLSI Design	3	0	2	4
	EC367	EC	Microprocessor and Microcontroller Lab	0	1	2	2
	EC369	EC	Microwave Engineering Lab	0	0	2	1
	HS301	HS	Entrepreneurship and IP Strategy	0	1	0	1
	DP391	PI	Design Project	0	0	0	3
				<b>12</b>	<b>2</b>	<b>12</b>	<b>23</b>

Semester - VI	-	EC	Program Elective (PE1)*	3	0	2	4
	-	EC	Program Elective (PE2)*	3	0	0	3
	-	EC	Program Elective (PE3)*	3	0	2	4
	-	EC	Program Elective (PE4)*	0	1	2	2
	-	EO	Elective from other Branch (EO1)	3	0	0	3
	-	OE	Open Elective (OE1)	3	0	0	3
				<b>15</b>	<b>1</b>	<b>6</b>	<b>19</b>
Semester - VII	-	EC	Program Elective (PE5)*	3	0	0	3
	-	EC	Program Elective (PE6)*	0	0	2	1
	-	EC	Program Elective (PE7)*	3	0	0	3
		EC	Program Elective (PE8)*	0	1	3	3
		EO	Elective from other Branch (EO2)	3	0	0	3
	-	OE	Open Elective (OE2)	3	0	0	3
	SI491	PI	Research/Industrial Internship	0	0	0	3
				<b>12</b>	<b>1</b>	<b>5</b>	<b>19</b>
Semester - VIII	BP490	PI	B.Tech. Project	0	0	36	12
				<b>0</b>	<b>0</b>	<b>36</b>	<b>12</b>

### \*Tentative List of Programme Electives for B.Tech. in ECE (PE1 - PE8)

S.N.	Programme Verticals	Courses
1	<b>Nanoelectronics/Nanotechnology</b>	Introduction to Nanoelectronics
		Nano Fabrication and Characterization Techniques
		Flexible and Organic Electronics
		Introduction MEMS and NEMS
		Semiconductor Device Modeling
2	<b>VLSI Design/Chip Design</b>	Analog IC Design
		Mixed-signal IC design
		Low Power VLSI Design
		VLSI Physical Design

		Physical Design Automation
		RF Integrated Circuit Design
3	System Design	Embedded Systems Design
		FPGA-Based System Design
		System-on-Chip (SoC) Design
		Digital System Design using Verilog/VHDL
		High-Level Synthesis (HLS) for FPGA
4	Testing and Verification	Advanced Topics in VLSI Testing
		VLSI Testing and Testability
		Digital System Verification
		Verification of SoC Designs
		Static Timing Analysis
5	AI Hardware & Neuromorphic Computing	AI Hardware Accelerators
		VLSI Design for Machine Learning
		In-Memory Computing
		Advanced Topics in Neuromorphic Computing
		FPGA-Based Neuromorphic Systems
6	AI/ML	Introduction to Artificial Intelligence
		Machine Learning
		Computer Vision
		Natural Language Processing
		Reinforcement Learning
		Optimization Techniques
7	Data Science	Introduction to Data Science
		Big Data Analytics

		Data Visualization and Analytics
		Programming for Data Science
<b>8</b>	<b>Signal Processing</b>	Adaptive Signal Processing
		Image and Video Processing
		Speech Signal Processing
		Advanced Digital Signal Processing
<b>9</b>	<b>RF and Microwave</b>	Microwave and mm-wave Circuits
		High Power mm/THz Wave Engineering
		Microwave Imaging
		Radar Signal Processing
		Thermal Imaging
		Introduction to Drone Technology
		Wireless Technologies: 5G and Beyond
		Advanced Wireless Communication Systems

## 6. B.Tech. in Information Technology

Semester	Course Code	Domain	Course Name	L	T	P	C
Semester - III	MA201	MA	Mathematics-III	3	1	0	4
	HS201	SE	Technical Writing	1	1	2	3
	CS201	CS	Object-Oriented Design & Programming	3	0	0	3
	CS261	CS	Object-Oriented Design & Programming Lab	0	0	2	1
	CS203	CS	Design and Analysis of Algorithms	3	0	0	3
	CS263	CS	Design and Analysis of Algorithms Lab	0	0	2	1
	EC201	EC	Digital Logic Design	3	0	0	3
	EC261	EC	Digital Logic Design Lab	0	0	2	1
	SC201	HS	Environmental Science	2	0	0	2
				<b>15</b>	<b>2</b>	<b>8</b>	<b>21</b>
Semester - IV	HS202	HS	Economics	3	0	0	3
	IT202	CS	Web Technology	3	0	0	3
	IT262	CS	Web Technology Lab	0	0	2	1
	IT204	CS	Database Management System	3	0	0	3
	IT264	CS	Database Management System Lab	0	0	2	1
	IT206	CS	Software Engineering	2	0	0	2
	IT264	CS	Software Engineering Lab	0	1	2	2
	IT208	CS	Computer Organization and Architecture	3	0	0	3
	IT268	CS	Computer Organization and Architecture Lab	0	0	2	1
				<b>14</b>	<b>1</b>	<b>8</b>	<b>19</b>
Semester - V	IT301	CS	Computer Networks	3	0	0	3
	IT361	CS	Computer Networks Lab	0	0	2	1
	IT303	CS	Operating Systems	3	0	0	3
	IT363	CS	Operating Systems Lab	0	0	2	1
	IT305	CS	Information Security	3	0	0	3
	IT365	CS	Information Security Lab	0	0	2	1
	IT307	CS	Software Project Management	2	0	0	2
	IT367	CS	Software Project Management Lab	0	1	2	2
	IT309	CS	Management Information System	3	0	0	3
	HS301	HS	Entrepreneurship and IP Strategy	0	1	0	1
	DP391	PI	Design Project	0	0	0	3

				<b>14</b>	<b>2</b>	<b>8</b>	<b>23</b>
Semester - VI	IT302	CS	Cloud System Management	2	0	0	2
	IT362	CS	Cloud System Management Lab	0	0	3	2
	-	CS	Program Elective (PE1)*	3	0	2	4
	-	CS	Program Elective (PE2)*	3	0	0	3
	-	CS	Program Elective (PE3)*	0	1	2	2
	-	EO	Elective from other Branch (EO1)	3	0	0	3
	-	OE	Open Elective (OE1)	3	0	0	3
				<b>14</b>	<b>1</b>	<b>7</b>	<b>19</b>
Semester - VII	-	CS	Program Elective (PE4)*	3	0	2	4
	-	CS	Program Elective (PE5)*	3	0	0	3
	-	CS	Program Elective (PE6)*	3	0	2	4
	-	EO	Elective from other Branch (EO2)	3	0	0	3
	-	OE	Open Elective (OE2)	3	0	0	3
	SI491	PI	Research/Industrial Internship	0	0	0	3
				<b>15</b>	<b>0</b>	<b>4</b>	<b>20</b>
Semester - VIII	BP490	PI	B. Tech. Project	0	0	36	12
				<b>0</b>	<b>0</b>	<b>36</b>	<b>12</b>

**\*Programme Electives for B.Tech. in IT (PE1 - PE6) is provided in Appendix A.**

## 7. B.Tech. in Mathematics and Computing

Semester	Course Code	Domain	Course Name	L	T	P	C
Semester - III	MA201	MA	Mathematics-III	3	1	0	4
	HS201	SE	Technical Writing	1	1	2	3
	MC201	MA	Calculus of Several Variables	3	1	0	4
	CS201	CS	Object-Oriented Design & Programming	3	0	0	3
	CS261	CS	Object-Oriented Design & Programming Lab	0	0	2	1
	CS203	CS	Design and Analysis of Algorithms	3	0	0	3
	CS263	CS	Design and Analysis of Algorithms Lab	0	0	2	1
	SC201	HS	Environmental Science	2	0	0	2
				<b>15</b>	<b>3</b>	<b>6</b>	<b>21</b>
Semester - IV	MC202	MA	Scientific Computing	2	1	2	4
	HS202	HS	Economics	3	0	0	3
	MC204	MA	Optimization Techniques	3	1	0	4
	MC206	CS	Machine Learning	3	0	2	4
	CS204	CS	Database Management System	3	0	0	3
	CS262	CS	Database Management System Lab	0	0	2	1
				<b>14</b>	<b>2</b>	<b>6</b>	<b>19</b>
Semester - V	MC301	MA	Advanced Statistics	3	1	0	4
	MC303	MA	Graph Theory	3	0	0	3
	MC305	MA	Modern Algebra and Cryptography	3	0	0	3
	MC365	MA	Modern Algebra and Cryptography Lab	0	0	2	1
	CS303	CS	Operating Systems	3	0	0	3
	CS363	CS	Operating Systems Lab	0	0	2	1
	CS307	CS	Artificial Intelligence	3	0	0	3
	CS367	CS	Artificial Intelligence Lab	0	0	2	1
	HS301	HS	Entrepreneurship and IP Strategy	0	1	0	1
	DP391	PI	Design Project	0	0	0	3
				<b>15</b>	<b>2</b>	<b>6</b>	<b>23</b>
Semester - VI	-	PH	Program Elective (PE1)*	3	1	0	4
	-	PH	Program Elective (PE2)*	3	1	0	4
	-	PH	Program Elective (PE3)*	3	0	2	4
	-	PH	Program Elective (PE4)*	0	1	2	2

	-	EO	Elective from other Branch (EO1)	3	0	2	4
	-	OE	Open Elective (OE1)	3	0	0	3
				<b>15</b>	<b>4</b>	<b>6</b>	<b>21</b>
Semester - VII	PH401	PH	Program Elective (PE5)*	3	1	0	4
	CS401	EO	Program Elective (PE6)*	3	1	0	4
	CS461	EO	Program Elective (PE7)*	3	0	2	4
	-	EO	Elective from other Branch (EO2)	3	0	2	4
	-	OE	Open Elective (OE2)	3	0	0	3
	SI491	PI	Research/Industrial Internship	0	0	0	3
				<b>15</b>	<b>2</b>	<b>4</b>	<b>22</b>
Semester - VIII	BP490	PI	B. Tech. Project	0	0	36	12
				<b>0</b>	<b>0</b>	<b>36</b>	<b>12</b>

### \*Tentative List of Programme Electives for B.Tech. in MAC (PE1 - PE7)

Electives	Mathematics Courses	Computer Science Courses
PE1	Complex Analysis	-
PE2	Dynamical Systems, Computational Fluid Dynamics	Coding Theory
PE3	Computational Algebra, Computational Number Theory	Automata Theory
PE4	Applied Linear Algebra	Computer Aided Design
PE5	Mathematics of Finance	Theory of Computation
PE6	Functional Analysis/ Topology	Image Processing, Big Data Analytics, Cloud computing, Information Retrieval, Reinforcement Learning
PE7	Computational Number Theory	Distributed and Parallel Computing
OE1 (Open Elective)	Climate Modelling and Data Assimilation, Quantum Computing	
OE2 (Open Elective)	Computational Physics	
EO1 (elective from other branch)	Computer Architecture	
EO2 (elective from other branch)	High Performance computing, Deep Learning	

**Programme Verticals for B.Tech. in MAC**

<b>Algebra and Cryptography</b>	<b>Analysis</b>	<b>Scientific Computing</b>
Galois theory	Real and Complex Analysis	Numerics of PDE
Finite fields and Coding theory	Functional analysis and Measure theory	Finite Element Methods
Quantum computing and Post Quantum Cryptography	Operator theory	Fourier Transform and Wavelet Analysis

---

## Semester-I

### Courses and Contents

Course Code	Course Name	L-T-P: C
MA101	Mathematics-I (Linear Algebra and Matrices)	3-1-0: 4

#### Objectives:

- To understand fundamental concepts of matrices, determinants, rank, and the solution of linear systems.
- To learn canonical matrix factorizations such as eigen decomposition, diagonalization, Schur, QR, and SVD.
- To comprehend the structure of vector spaces, linear transformations, and their canonical forms.
- To apply numerical methods for solving linear systems and computing eigenvalues and eigenvectors.

**Learning Outcomes:** Upon successful completion of this course, student will be able to:

- Perform matrix operations and analyze linear systems using concepts of rank and determinants.
- Apply various matrix factorizations to study properties of matrices and linear transformations.
- Demonstrate understanding of vector spaces, basis, dimension, and linear mappings in abstract and computational settings.
- Implement and analyze numerical algorithms for solving linear systems and computing eigenvalues and eigenvectors.

**Prerequisite:** None

#### Contents:

**Module I:** Vector Spaces: Vector Space over the set real numbers (Field), Linear Independence of Vectors, Bases in a Vector Space, Dimension of a Vector Space, Direct Sum Decomposition of a Vector Space, Linear Transformations, Change of Bases, Rank of a Linear Transformation.

**Module II:** Matrices and Linear Systems: Matrix operations (addition, multiplication), Block-Partitioned Matrices and Block Operations, Elementary Row and Column Operations, Determinant of a Square Matrix, Properties of the Determinant Function, Cofactor Expansion, Rank of a Matrix, The System of Linear Equations:  $Ax=b$

**Module III:** Canonical Factorizations: Eigenvalues and Eigenvectors, Companion Matrices and Characteristic Polynomial, diagonalization-Matrices with a full-set of

eigenvectors, The Cayley-Hamilton Theorem, unitary diagonalization of a symmetric matrix, Inner product spaces, Gram-Schmidt process, projections and least squares approximation, QR-Decomposition, Singular Value Decomposition.

**Module IV:** Numerical methods: Iterative methods (Jacobi, Gauss-Seidel, Relaxation) for solving linear systems, computing of eigenvalues and eigenvectors.

#### Text Books:

- Linear Algebra and Its Applications. David C. Lay, Steven R. Lay, Judith Joanne McDonald. 5th ed, Pearson ed, 2020

#### Reference Books:

- Linear Algebra, Kunze Ray, Hoffman Kenneth 2nd ed, Phi Learning, 2014.
- Fundamentals of Matrix Computations, David S. Watkins, 3rd ed, Wiley.
- Introduction to Linear Algebra, Gilbert Strang, 5th ed, SIAM, 2016.

Course Code	Course Name	L-T-P: C
MC101	Combinatorial Linear Algebra	1-0-0: 1

#### Objectives:

- Introduce students to the interplay between linear algebra and combinatorics through concrete, problem-driven examples.
- Develop the ability to model discrete problems using vectors, matrices, and linear transformations.
- Expose students to powerful linear-algebraic techniques such as rank arguments, eigenvalues, inner products, and spectral methods in combinatorial settings.
- Familiarize students with classical and modern applications including error-correcting codes, graph theory, geometry, and complexity.

**Learning Outcomes:** Upon successful completion of this course, student will be able to:

- Apply linear algebraic methods to solve combinatorial problems involving sets, graphs, and distances.
- Use spectral techniques and eigenvalues to study graphs, including connectivity, cuts, and spanning trees
- Model and reason about error-correcting codes using vector spaces over finite fields.
- Design and analyze randomized and deterministic algorithms, such as efficient matrix multiplication checking

- Communicate mathematical ideas clearly by translating combinatorial questions into linear-algebraic formulations and proofs

**Prerequisite:** None

**Contents:**

The following topics from the book *Thirty-three Miniatures: Mathematical and Algorithmic Applications of Linear Algebra* by Jiří Matoušek will be discussed in the course.

- Fibonacci Numbers, The clubs of odd town, Same size Intersections.
- Error Correcting codes.
- Odd distances, Are these distances Euclidean.
- Packing complete bipartite graphs.
- Equiangular lines, Where is the triangle.
- Checking matrix multiplication.
- Medium-size Intersection is Hard to Avoid, The End of the Small Coins.
- Walking in the Yard.
- Counting Spanning Trees.
- Cutting Cheaply Using Eigenvectors

**Text Book:**

- Thirty-three Miniatures: Mathematical and Algorithmic Applications of Linear Algebra*. J. Matoušek, AMS, 2010.

Course Code	Course Name	L-T-P: C
PH101	Physics-I (Classical and Modern Physics)	3-1-0: 4

**Objectives:** This course provides engineering students with important foundational knowledge about the physical systems, and applications to sectors such as healthcare, energy, communication, agriculture, and computation. The course also includes weekly small-group problem-solving tutorial sessions.

**Learning Outcomes:** On successful completion of this course, students should be able to:

- Demonstrate knowledge of the physical principles that describe classical and quantum mechanics.
- Use the methods of algebra, vectors, and calculus in quantitative and qualitative predictions of behavior of physical systems.
- Associate the correct unit with the different physical quantities they use.
- Apply physics principles to areas such as quantum computation, communication, energy, agriculture, and healthcare.

**Prerequisite:** None

**Contents:**

**Module I** Coordinate Systems: Cartesian, cylindrical and spherical coordinate systems, Inertial and non-inertial frames of reference, Centrifugal and Coriolis forces.

**Module II** Motion under Central Force: Concept of Central Force and its universal features, Classification of orbits and Kepler's problem, scattering in central force.

**Module III** Oscillatory Motion: Free, Damped, and forced oscillatory motion, resonance, Beyond the small oscillation approximation.

**Module IV** Foundations of Quantum Mechanics: Black body radiation, Planck's hypothesis, Compton effect, Photoelectric effect: Einstein's quanta, Bohr's semiclassical atomic model, Young's Double-slit experiment, and de-Broglie's hypothesis, Davisson and Germer experiment, Heisenberg Uncertainty Principle.

**Module V** Machinery of Quantum Mechanics: Postulates of quantum mechanics, Schrödinger Equation, Observables, eigenvalue and eigenfunctions. Applications of one-dimensional Schrödinger Equation: Particle in a box, Finite Potential well, Harmonic Oscillator, and Hydrogen atom.

**Module VI** Topics on Applied Physics: Healthcare (NMR, MRI), Sources of Renewable Energy, Communication (Fiber Optics), Agrophysics, Quantum computation.

**Text Books:**

- An Introduction to Mechanics*; D. Kleppner and R. Kolenkow, Second Edition.
- Concepts of Modern Physics*; A. Beiser, Sixth Edition.
- Quantum Mechanics Concepts and Applications*, Nouredine Zettili, Second Edition, Wiley 2009.
- Frank S. Crawford, Berkeley Physics Course Vol 3: Waves and Oscillations*, McGraw Hill, 1966.
- Introduction to Electrodynamics* by David J. Griffiths, Fourth Edition.

Course Code	Course Name	L-T-P: C
PH161	Experimental Physics-I	0-0-2: 1

**Objectives:** The laboratory course is designed to give experimental exposure to engineering students about the important foundational knowledge about mechanics (both classical and quantum), and their applications to common engineering systems.

**Learning Outcomes:** At the end of the course, the students shall be able to acquire the following learning outcomes:

Knowledge: The student has gained knowledge about

1. Motion of bodies in inertial and non-inertial frames, central forces, oscillations & its types.
2. Planck's constant and Quantization of light, Franck-Hertz experiments and spectral lines.
3. Nature of light and its properties.

General competence: The student has gained

1. Increased ability for reflection and insight in how physics connects cause and effect for simple processes in nature.
2. Increased understanding on how basic concepts and methods are the foundations of more advanced fields.
3. Increased capability to recognize the basic concepts of physics at play in everyday life phenomena.
4. Training in analyzing complex problems by decomposing them into simpler tasks.

**List of recommended experiments:**

1. Perform Elastic and Inelastic collisions between two different masses on a linear air track and verify conservation of momentum and energy.
2. Analyzing damped oscillation using the spring-mass system in a different medium. Determination of damping constant by varying mass of the spring-mass system.
3. Study of photoelectric effect and determination of Planck's constant, and to verify inverse square law of radiation using a photoelectric cell.
4. Study of the centripetal acceleration.
5. Determination of Rydberg's constant and analyzing the Balmer series of hydrogen spectra.
6. Determination of  $e/m$  ratio using Helmholtz coil.
7. Measurement of wavelength of light source using Michelson Interferometer.
8. Newton rings experiment for refractive index measurement.
9. Diffraction grating experiment with spectrometer.
10. Measurement of angle of minimum deviation through dispersion of light from a prism spectrometer.

**Reference:** Laboratory Manual

Course Code	Course Name	L-T-P: C
CS101	Computer Programming and Problem Solving	3-0-0: 3

**Objectives:** This course aims to provide a foundational understanding of computer programming and its role in problem-solving. Students will learn to develop well-structured programs using the C language and implement basic data structures to enhance computational efficiency.

**Learning Outcomes:** On successful completion of this course, students should be able to:

1. Problem solving through computer programming.
2. Understand basic problem solving and programming concepts.
3. Visualize and model the basic real-life problems into computational problems.
4. Ability to deal with different input/output methods.
5. Ability to use different data structures.
6. Understand the basic computing environment, limitations, and usage.

**Prerequisite:** None

**Contents:**

**Module I:** Introduction to computers, basic organization, problem-solving techniques, debugging, programming language aspects (syntax, semantics). Overview of system and application software, compilers, and interpreters. Basics of C: data types, storage classes, operators, constants, I/O functions, integer arithmetic, and typecasting.

**Module II:** Control structures (if-else, switch, loops), problem-solving using loops (GCD, sum, max, deviation, trigonometric functions). Introduction to 1D arrays and strings, string operations (e.g., palindrome check).

**Module III:** Problem-solving techniques: sorting, partitioning, merging arrays, computing square roots, recursion (factorial, Fibonacci), dynamic programming, nested loops, bitwise operations, macros, and preprocessor usage.

**Module IV:** Structures (struct, typedef), functions (call by value/reference), function usage in large power computations, multi-dimensional arrays (matrix operations), command-line arguments, and variable argument functions.

**Module V:** Pointers (arithmetic, arrays vs. pointers, malloc()), linked lists, function pointers, enum, and file handling (read, write, append).

**Text Books:**

1. Let us C, Yashavant Kanetkar, Latest Edition, BPB Publication.
2. The C Programming Language, Brian W. Kernighan and Dennis Ritchie, Latest Edition, Prentice Hall.

**Reference Books:**

1. Programming in ANSI C, E. Balagurusamy, Latest Edition, McGraw Hill.
2. Education Solutions Limited, I. T. L. (2004). Introduction to Computer Science. India: Pearson Education.
3. How to Solve it by Computer, RG Dromey, PHI.

Course Code	Course Name	L-T-P: C
IT161	Computer Programming and Problem Solving Lab	0-0-3: 2

**Objectives:** This course aims to enhance problem-solving and analytical skills while developing proficiency in implementing programs using C or Python.

**Learning Outcomes:** On successful completion of this course, students should be able to:

1. Identify and model computational tasks for a given problem.
2. Develop algorithms or flowcharts and implement them in a chosen programming environment.
3. Select appropriate data representations based on problem requirements.
4. Execute programs, identify and resolve errors, and analyze the execution environment.
5. Collaborate effectively as a team member.

**List of Experiments:**

1. Write a program to take user input and display different data types (int, char, float, double). Implement type casting and the sizeof() function.
2. Implement arithmetic, relational, logical, and bitwise operations in C. Demonstrate operator precedence.
3. Write a program to find the maximum of three numbers using if-else and the ternary operator.
4. Implement a program to compute the sum, maximum, and standard deviation of N numbers using for, while, and do-while loops.
5. Implement string functions such as length, copy, concatenation, and checking if a string is a palindrome without using built-in functions.
6. Implement bubble sort and binary search on an array of integers.
7. Compute factorial, Fibonacci series, and square root of a number using recursion and compare it with an iterative approach.
8. Use bitwise operators to check the membership of an element in a set and perform union/intersection operations.

9. Implement a structure to store student records and perform operations using functions (passing by value and reference).
10. Implement dynamic memory allocation using malloc() and free(). Demonstrate pointer arithmetic and pointer to structures.
11. Write a program to read, write, and append data to a text file.
12. Implement a calculator program that takes input from the command line and performs operations using function pointers.

**Reference:** Laboratory instructions and handouts.

Course Code	Course Name	L-T-P: C
EC101	Basic Electronics-I	3-1-0: 4

**Objectives:** The course presents the principles of analog circuit analysis and design. It introduces the basic concepts and characteristics of the electronic devices and circuits. The tutorials help to develop the ability of analyzing actual electronic circuits that implement the basic circuits presented.

**Learning Outcomes:** On successful completion of this course, students should be able to:

1. Characterize circuit elements, sources, semiconductors, diodes, transistors, oscillators and operational amplifiers.
2. Design application oriented simple analog circuits.

**Prerequisite:** None

**Contents:**

**Module-I:** Fundamentals of Semiconductors: Energy bands, direct and indirect band-gap semiconductors, intrinsic and extrinsic semiconductors, equilibrium charge-carrier concentration, charge-carrier density and distribution, Fermi level. Charge-carrier transport: Diffusion current, drift current, mobility and resistivity, generation and recombination of carriers, Poisson and continuity equations.

**Module-II:** Diodes: p-n junction, Zener diode, I-V Characteristics, Diode Models, Rectifiers and Voltage Regulators, Clippers and Clampers, Introduction to Special Purpose Diodes: Varactor Diode, LEDs, Solar Cells, Photo-diodes, Tunnel Diode, Schottky Diode.

**Module-III:** Basics of Bipolar Junction Transistors (BJTs): BJT structure, Basic BJT operation mechanism, Input and Output characteristics of common-emitter configuration, common-base configuration, common collector configuration, and current amplification factors.

**Module-IV:** Basics of Field Effect Transistors (FETs): JFETs-Device structure, I-V Curves, Transconductance Curve, MOS capacitor, MOSFETs-Device structure, I-V Curves, Depletion-Mode, Enhancement-Mode, MOSFET as a switch, CMOS, Power FETs.

**Module-V:** New and Emerging Semiconductors and their Devices: Graphene (2D), Carbon Nano Tubes (1D), Quantum Dots (0D), and other related novel materials and devices.

**Text Books:**

1. *Physics of Semiconductor Devices*, S.M. Sze, 2nd Edition, Wiley.
2. *Electronic Devices*; 9th Edition, Thomas L. Floyd, Pearson.
3. *Electronic Principles*, 9th Ed, Albert Malvino, and David Bates, Tata McGraw-Hill, 2021.
4. *Solid State Electronic Devices*, Streetman, B. G., Banerjee, S, 7th Edition, PHI.
5. *Microelectronic Circuits: Theory and Applications*, A.S. Sedra and K.C. Smith, Sixth Edition, Oxford University Press.

Course Code	Course Name	L-T-P: C
EC161	Basic Electronics Lab-I	0-0-3: 2

**Objectives:** The objective of the laboratory is to provide experimental hand-on experiences to the topics covering the course ‘Basic Electronics-I’.

**Learning Outcomes:** On successful completion of this laboratory course, students should be able to:

1. Design and construct circuits to verify theoretical relationships involving devices and circuits.
2. Familiar with basic electrical measurements using laboratory instruments.
3. Able to design, assemble, build and test simple electronic circuits.

**List of Experiments:**

1. Introduction to circuit elements and basic equipments: Resistors, Capacitors, Inductors, Diodes, Transistors, Oscilloscope, Function generator, Power supply, Cables and Switches.
2. Transient- and steady-state response of RC circuits and design RC Filters.
3. Current-voltage characteristics of a *p-n* junction and Zener diode at room temperature.
4. To study the applications of *p-n* diode as Half- and Full-Wave Rectifiers.
5. Design a regulated power supply using Zener diode and verify its characteristics.
6. To design and analyse clipper circuits using *p-n* diodes.

7. To design and analyse clamper circuits using *p-n* diodes.
8. Input and output current-voltage characteristics of *n-p-n* bipolar junction transistors in common-emitter configuration and determine transistor parameters.
9. To understand the characteristics of a Light Dependent Resistor (LDR) and use it to design a light dependent electronic switch.
10. To study and design MOSFET (*n*-Channel and *p*-Channel) as a switch.

**Reference:** Laboratory Manual

Course Code	Course Name	L-T-P: C
EC103	Introduction to ECE	1-0-0: 1

**Objectives:** The course objective is to introduce students to the fundamental concepts and principles of electronics, communication systems, and their practical applications in modern technology. The course provides a broad understanding of how electronic circuits, signals, and communication systems operate, preparing students for more specialized topics in electronics, telecommunications, and signal processing. The goal is to equip students with the necessary foundation in the theory and practice of electronics and communication systems.

**Learning Outcomes:** On successful completion of this course, students should be able to:

1. Get a basic idea of electronic components and circuits.
2. Get a basic idea of communication Systems.
3. Get a basic idea of digital electronics and microcontrollers.
4. Get a basic idea of VLSI Design and Artificial Intelligence.

**Prerequisite:** None

**Contents:**

**Module-I:** Overview of Digital and Analog Electronic Components and Circuits.

**Module-II:** Overview Communication Systems.

**Module-III:** Overview of Microprocessor and Microcontrollers.

**Module-IV:** Overview of Specialization of VLSI Design and Artificial Intelligence.

**Reference:** Will be prescribed by the Course Instructor.

Course Code	Course Name	L-T-P: C
CC101	Cybersecurity Essentials	3-0-0: 3

**Course Objectives:** The course objective is to provide a fundamental understanding of all cybersecurity domains and learn the challenges and impacts of threats.

**Learning Outcomes:** On successful completion of the course students will be able to demonstrate:

1. Assess and evaluate an organization's cybersecurity requirements.
2. Understand cybersecurity regulations and the roles of international law.
3. Create and implement security architecture for the organization.
4. Comprehend fundamental concepts regarding data privacy and associated attacks.

**Prerequisites:** None

**Contents:**

**Module I:** Cyber Security Fundamentals: Introduction to Cybersecurity, Issues in cybersecurity, Challenges of cyber security, Essentials of Security, Impacts due to lack of security, Security Model Work, Confidentiality, Integrity, and Availability (CIA) Triad, Types of Security Threats, Hackers, Types of hackers.

**Module II:** Network and Security Concepts: Information Assurance Fundamentals, Basic Cryptography, Symmetric Encryption, Public Key Encryption, Domain Name System (DNS), Firewalls, Virtualization.

**Module III:** Attacker Techniques and Exploitation: Anti-Forensics. Proxies, Tunneling Techniques, Fraud Techniques, Phishing, Smishing, Vishing, and Mobile Malicious Code, Rogue Antivirus, Click Fraud, Exploitation: Techniques to Gain a Foothold, DoS Conditions, Brute Force and Dictionary Attacks, Distributed denial of service (DDoS), Man-in-the-middle attack (MitM), Misdirection, Reconnaissance, and Disruption Methods.

**Module IV:** Detection and Prevention of Malicious Codes: Self-Replicating Malicious Code, Evading Detection and Elevating Privileges, Rootkits, Spyware, Attacks against Privileged User Accounts and Escalation of Privileges, Token Kidnapping, Virtual Machine Detection, Stealing Information and Exploitation.

**Module V:** Defense and Analysis Techniques: Memory Forensics, Capabilities of Memory Forensics, Memory Analysis Frameworks, Dumping Physical Memory, Installing and Using Volatility, Finding Hidden Processes, Volatility Analyst Pack.

**Text Books:**

1. Charles J. Brooks, Christopher Grow, Philip Craig, Donald Short - Cybersecurity Essentials, October 2018, Sybex publications, ISBN: 9781119362395
2. James Graham, Ryan Olson, Rick Howard - Cyber Security Essentials Paperback, Auerbach Publications (O'reilly), 1st edition (15 December 2010), ISBN-10 : 1439851239, ISBN-13 : 978-1439851234.
3. Cyber Security Understanding Cyber Crimes, Computer Forensics and Legal Perspectives by Nina Godbole and Sunit Belpure, Publication Wiley
4. Anti-Hacker Tool Kit (Indian Edition) by Mike Shema, Publication McGraw Hill.

**Reference Books:**

1. Cyber Security Essentials, James Graham, Richard Howard and Ryan Olson, CRC Press.
2. Introduction to Cyber Security, Chwan-Hwa(john) Wu, J. David Irwin, CRC Press T&F Group.

Course Code	Course Name	L-T-P: C
CC103	System Administration and Maintenance	0-1-2: 2

**Course Objectives:** The course aims to equip students with essential skills in Linux system administration, including command-line operations, user and security management, task automation, system monitoring, networking utilities, and software and backup management for effective system maintenance.

**Learning Outcomes:** By the end of the course students will be able to:

1. Perform Linux command-line operations for file and system management.
2. Manage users, groups, and permissions with secure configurations.
3. Develop shell scripts to automate routine administrative tasks.
4. Monitor system performance and analyze logs for troubleshooting.
5. Use networking tools and implement secure remote access.
6. Manage software packages, version control, and backup and recovery processes.

**Prerequisites:** None

**Contents:**

**Module I:** Linux Environment and Command-Line Practice Introduction to Linux environments and using terminal. Essential command-line operations: file

system navigation, file and directory manipulation, text processing utilities, and understanding file permissions and ownership. Basic process monitoring and management using commands such as ps, top, and kill, interpreting system state.

**Module II:** User Administration and Secure Configuration, User and group management, password setting, and access control. Permission models (chmod, chown, umask). Use of sudo for privilege escalation. Managing services, enforcing password policies, and securing user access.

**Module III:** Shell Scripting and Task Automation Bash scripting using variables, conditional statements, loops, and functions to automate routine administrative tasks. Automated job scheduling using cron. Log inspection and system maintenance.

**Module IV:** System Monitoring, Logging, and Network Utilities System monitoring tools such as top, htop, vmstat, and iostat. Log file analysis, and journaling systems. Networking tools including ping, netstat/ss, and traceroute for basic network diagnostics. Secure remote access using SSH and basic firewall configuration through terminal.

**Module V:** Software Management, Version Control, and Backup Practices Installation and management of software using package managers (apt/yum). Version control using Git and collaboration basics via GitHub. Backup and recovery techniques using tar and rsync.

**Text Books:**

1. Text Books: "The Linux Command Line: A Complete Introduction" by William Shotts
2. "Pro Git" by Scott Chacon and Ben Straub.

Course Code	Course Name	L-T-P: C
CCS105	Digital Design and Computer Architecture	3-0-2: 4

**Course Objectives:** To study the fundamentals of digital logic design and computer architecture of digital computers (CPU, memory, I/O, software, pipelining and parallelism). Discussions will include digital logic, microprogramming and performance enhancement of processors. Such knowledge leads to better understanding computer organization and architecture, can be used in the design and implementation of computer systems or as foundation for more advanced computer-related studies.

**Learning Outcomes:** On successful completion of the course students will be able to:

1. Understand the fundamentals of digital logic design implementation of digital circuits.

2. Get hands-on experience of software and hardware.
3. Familiarize with pipeline architecture and instruction level parallelism.
4. Understand dynamic instruction scheduling and thread level parallelism.
5. Familiarize themselves with the memory system and its interaction with the processor.

**Prerequisite:** None

**Contents:**

**Module I:** Number systems and Boolean algebra: Different number systems, conversion between them, binary arithmetic, complement representation, floating-point representation, logic gates, Boolean algebra laws, De Morgan’s theorem, Sum of Products (SOP) and Product of Sums (POS).

**Module II:** Combinational and Sequential logic: logic minimization using K-map method, QuineMc Cluskey method, multiplexers/ demultiplexers, decoders/ encoders, adders, ripple carry adder, carry-lookahead adder, multiplication - shift-and-add, Booth multiplier. Simple circuits with feedback, basic latches, clocks, Flip flops, shift register, synchronous and asynchronous counters, Finite State Machine (Moore/Mealy Machines).

**Module III:** Von Neumann: Functional units, ALU, data paths architecture, registers, instruction set architecture (ISA), addressing modes.

**Module IV - Pipelining and Parallelism:** An overview of pipelining, throughput and speedup, pipelined data path and control, data dependency and hazard, control hazard and structural hazard, instruction level parallelism (ILP) concepts and challenges, basic compiler techniques for exposing ILP, ILP using dynamic scheduling, VLIW, superscalar architecture, overview of thread level parallelism.

**Module V:** Memory system and I/O: Principles of temporal and spatial locality; cache memories (address mapping, block size, replacement and store policy); virtual memory (page table, TLB); disk organization and data access from disk drive, programmed I/O, interrupt-driven I/O, Handshaking, DMA, interrupts.

**Text Books:**

1. Digital Design, Morris Mano, Prentice Hall, 2002.
2. Digital Fundamentals, 10 th Ed, Floyd T L, Prentice Hall, 2009.
3. Computer Architecture: A Quantitative Approach, 5th Edition By John L Hennessy; David A Patterson.

**Reference Books:**

1. Digital electronics: principles, devices and applications, Albert Paul Malvino and Jerald A. Brown, John Wiley & Sons, 2007.
2. Computer system architecture, Morris Mano, revised third edition, Pearson.
3. Computer Organization and Architecture, 8th Ed, Stallings, Pearson, 2010.

#### Tentative List of Lab Experiments:

1. Design and implementation of data path architecture and program counter.
2. Design and implementation of arithmetic logic unit (ALU).
3. Design and implementation of ripple carry adder and carry-look-ahead adder.
4. Design and implementation of Booth's multiplier and combinational multipliers.
5. To understand the basic principles of pipeline design for single instruction, including the problems of data and branch hazards.
6. To understand pipeline design for multiple instructions, including the problems of data and branch hazards.
7. To understand the multiple instructions (beq, lw and st) working using pipelined processors.
8. To understand the arithmetic instruction (add, mult, div, sub) working using instruction level parallelism.
9. To understand the implementation of instruction level parallelism using Scoreboard algorithm.
10. To understand the implementation of instruction level parallelism using the Tomasulo algorithm.

#### Text Books:

1. Digital Fundamentals, 10th Ed, Floyd T L, Prentice Hall, 2009.
2. Computer Architecture: A Quantitative Approach, 5th Edition By John L Hennessy & David A Patterson.

#### Reference Books:

1. Computer system architecture, Morris Mano, revised third edition, Pearson.
2. Computer Organization and Architecture, 8th Ed, Stallings, Pearson, 2010.

Course Code	Course Name	L-T-P: C
AI101	Introduction to Artificial Intelligence	2-0-0:2

**Description:** The course aims to provide a foundational understanding of Artificial Intelligence (AI) concepts, techniques, and applications.

**Objectives:** The course aims to: Introduce the fundamental concepts, history, and scope of Artificial Intelligence (AI). Familiarize students with the current status, key tools, and interdisciplinary nature of AI. Provide an understanding of the mathematical foundations that underpin AI techniques. Develop problem-solving skills using state space search, local search, and planning methods. Expose students to practical applications of AI in vision and natural language processing. Encourage critical reflection through book reviews and discussions on the role of AI in society.

**Learning Outcomes:** On successful completion of this course, students will be able to:

**Knowledge and Understanding:** Explain the history and evolution of AI, including current trends and challenges. Identify and describe the interdisciplinary fields that contribute to AI development. Summarize the role of mathematics (linear algebra, probability, optimization) in AI.

**Skills and Application:** Formulate and solve problems using state space search and local search algorithms. Apply basic planning and logic-based reasoning to structured problems. Illustrate core applications of AI in vision and NLP with real-world examples.

**Critical Thinking and Reflection** Evaluate books or articles on AI from both technical and societal perspectives. Discuss ethical and societal implications of AI applications. Connect foundational AI methods with emerging trends in tools and technologies.

**Prerequisites:** None

**Contents:**

**Module I:** Foundations of AI, Introduction to AI: What is AI, Scope and Relevance, History of AI: Classical era, Expert Systems, ML & Deep Learning eras, Current Status of AI (Q1): Contemporary trends, breakthroughs, limitations, What all fields are involved in AI: CS, Math, Cognitive Science, Linguistics, Neuroscience, Robotics, Tools and Technologies: Python, TensorFlow, PyTorch, scikit-learn, data resources

**Module II:** Mathematics and Core Techniques: Mathematics and AI (Q2): Linear Algebra, Probability, Calculus, Optimization (overview), Logic Puzzles: Constraint satisfaction, rule-based reasoning, examples in AI

**Module III:** Applications and Reflections, Applications in Vision: Object detection, recognition, image

understanding, Applications in NLP: Text processing, language models, translation, chatbots

Book Review 1: Foundational AI literature (e.g., Russell & Norvig, Mitchell)

Book Review 2: Popular/critical book on AI and society/ethics

Wrap-up Discussion: Linking fundamentals, techniques, and applications

#### Text Books:

1. Stuart Russell, Peter Norvig, Artificial intelligence: A Modern Approach, Prentice Hall, Fourth edition, 2020.
2. Nils J. Nilsson, Artificial Intelligence: A New Synthesis, Morgan-Kaufmann, 1998.

#### Reference Books:

1. Judea Pearl, Heuristics: Intelligent Search Strategies for Computer Problem Solving, Addison-Wesley Publishing Company, 1984.
2. Biere, A., Heule, M., Van Maaren, H., Walsh, T., Handbook of Satisfiability, IOS Press, 2009.

Course Code	Course Name	L-T-P: C
HS101	Spoken and Written Communication	2-0-2: 3

**Objectives:** To develop students' communicative competence in English by enhancing spoken and written skills across listening, speaking, reading, and writing, with emphasis on verbal and non-verbal communication, phonetics, discourse, and comprehension. The course builds confidence for academic, social, and professional contexts while strengthening pronunciation, structure, style, and effective writing through focused practice and applied language use in diverse real-world situations.

#### Learning Outcomes:

1. Improve speaking ability in English in terms of fluency, clarity, and comprehensibility.
2. Attain competence in all four language skills: listening, speaking, reading, and writing (LSRW).
3. Understand the fundamentals of communication, including verbal and non-verbal elements.
4. Develop proficiency in articulatory phonetics and correct pronunciation.
5. Enhance listening skills through exposure to authentic spoken English.

6. Build confidence in speaking through both formal and informal discourse-based activities.
7. Review and apply grammatical forms of English in specific communicative contexts.
8. Strengthen writing skills through planning, drafting, revising, and editing structured compositions.
9. Improve reading comprehension and analytical abilities.
10. Develop public speaking and interpersonal communication skills.
11. Increase self-awareness of language use and appreciate culturally-bound conventions of academic English writing.

#### Prerequisite:

Basic understanding of English language (spoken and written).

#### Contents:

**Module-I:** Fundamentals of Communication: Elements of communication: sender, receiver, message, channel, noise, feedback, and barriers. Importance of English language for effective communication. Communication types: verbal, non-verbal, formal, and informal communication. Audience analysis, purpose, and information gap principle. General and business communication.

**Module II:** English for Specific Purposes: Technical vocabulary, Acronyms, Initialisms, Abbreviations, Words often confused, Synonyms, Antonyms, Collocations, Workplace communication, Contextual language use, Professional communication skills, Preparation for real-world tasks such as participating in meetings, presenting, writing reports

**Module-III:** Phonetics and Pronunciation: Introduction to Articulatory Phonetics. International Phonetic Alphabet (IPA) symbols. Anatomy of speech organs and phonetic transcription. British English pronunciation norms. Correct pronunciation for dictionary usage and professional settings.

**Module-IV:** Spoken Communication and Public Speaking: Improving spoken English through cinema-based learning. Discourse practice: describing and analyzing film scenes. Class discussions and debates on contemporary issues. Confidence building exercises for effective verbal communication.

**Module-V:** Listening and Writing Skills: Exposure to TED Talks, radio news broadcasts, and public speeches. Active listening strategies and responding in writing. Writing exercises: short compositions (50-150

words). Contextual writing tasks based on campus life and classroom discussions.

**Module-VI: Reading Comprehension and Analytical Skills:** Focused reading of short stories: An Astrologer's Day, The Gift of the Magi, The Monkey's Paw, Genes, newspaper editorials, and essays. Identification of challenging vocabulary, idioms, and complex sentence structures. Application of reading strategies: skimming, scanning, intensive, and extensive reading. Reading comprehension assessments based on seen and unseen passages. Development of critical thinking and interpretative skills through analytical reading tasks.

**Textbooks:**

1. *Prism: Spoken and Written Communication, Prose & Poetry* published by Orient Longman, 2008.
2. Raman, Meenakshi, and Sangeeta Sharma. *Technical Communication: Principles and Practice*. 2nd ed., Oxford University Press, 2009.
3. Kumar, Sanjay, and Pushp Lata. *English Language and Communication Skills for Engineers*. Oxford University Press, 2018.
4. Tebeaux, Elizabeth, and Sam Dragga. *The Essentials of Technical Communication*. 4th ed., Oxford University Press, 2020.

**Reference Books:**

1. Dhanavel, S. P. *English and Communication Skills for Students of Science and Engineering*. Orient Blackswan, 2009.
2. Day, R. A. *Scientific English: A Guide for Scientists and Other Professionals*. 2nd ed., Universities Press, 2000.
3. Ladefoged, Peter. *A Course in Phonetics*. 5th ed., Harcourt, Brace, Jovanovich, 2006.

**Recommended Readings (Essays and Short Stories):**

1. *An Astrologer's Day* - R. K. Narayan
2. *The Gift of the Magi* - O. Henry
3. *The Monkey's Paw* - W.W. Jacobs
4. *Genes* - Sudha Murthy

Course Code	Course Name	L-T-P:C
HS103	Artificial Intelligence and Ethics	2-0-0:2

**Description:** This course delves into the intersection of Artificial Intelligence and ethical principles, providing students with the tools to analyze and address the ethical challenges posed by AI technologies. The course

covers foundational ethical theories and frameworks, examines key issues such as bias, fairness, privacy, accountability, and transparency, and highlights the societal and human rights implications of AI.

**Objectives:** This course aims to provide students with a comprehensive understanding of the ethical challenges and societal implications associated with the development and deployment of Artificial Intelligence. It explores foundational ethical principles, frameworks, and real-world dilemmas to address bias, fairness, privacy, transparency, and accountability in AI systems. The course emphasizes the societal impact of AI technologies, the role of regulatory frameworks, and the alignment of AI with human rights. Students will be equipped to critically evaluate and develop responsible AI solutions that prioritize ethical practices and inclusivity.

**Learning Outcomes:** Understand Ethical Principles and Frameworks in AI, Evaluate and Address Societal and Legal Impacts of AI, Develop Skills for Ethical AI Implementation

**Prerequisites:** None

**Contents:**

**Module I:** Foundations of AI Ethics: Introduction to AI Ethics, Importance of ethics in the development and deployment of AI, Overview of key ethical principles: Autonomy, fairness, beneficence, justice, and non-maleficence, Ethical Frameworks and Philosophies: Utilitarianism, deontology, and virtue ethics applied to AI. Ethical dilemmas in AI (e.g., trolley problem in autonomous vehicles).

**Module II:** Bias and fairness in AI systems: Types of bias (data bias, algorithmic bias, user bias). Examples of biased AI systems (e.g., biased hiring tools, facial recognition). Techniques to mitigate bias in AI models. Fairness and Equity: Definitions of fairness in AI decision-making. Trade-offs between fairness and accuracy in machine learning models.

**Module III:** Privacy, surveillance, and data ethics: Ethical concerns in data collection and use. Case studies on breaches of privacy in AI systems. Data minimization and anonymization techniques. Applications of AI in monitoring and surveillance. Balancing public safety and individual privacy rights.

**Module IV:** Accountability, transparency, and explainability: Who is responsible when AI systems fail? Legal and ethical implications of automated decision-making. Accountability and responsibility in the collection, analysis and use of data. Importance of interpretable AI models. Techniques to improve model Explainability (e.g., LIME, SHAP).

**Module V:** Societal impacts of AI: The impact of automation on jobs and industries. Preparing for an AI-driven economy. Addressing the digital divide and unequal access to AI technologies. Ensuring inclusivity in AI systems. Access to and control over information, governance of social media.

**Module VI:** Autonomous systems and ethical challenges: Decision-making algorithms and moral dilemmas (e.g., life-and-death decisions). Regulatory frameworks for autonomous systems. Ethical concerns in AI-powered weapons and surveillance. International agreements and frameworks for ethical use. Interacting with AI agents. Large Language Models and the Ethics of AI: Impact, Gaps and Opportunities

**Module VII:** AI and human rights: Ensuring AI aligns with human rights principles. Case studies on AI systems violating rights (e.g., discriminatory algorithms). Overview of AI ethics guidelines (e.g., DPDP act (India), GDPR (EU), EU AI Act, IEEE Ethics Framework). The role of governments and institutions in enforcing ethical AI.

**Module VIII:** Future of AI ethics and emerging trends: Ethical challenges in developing highly intelligent AI systems. Controlling risks associated with AI surpassing human intelligence. Future international policies for AI. Collaboration for ethical AI development.

**Text Books:**

1. Ethics of Artificial Intelligence by S. Matthew Liao, Oxford University Press, 2020.
2. Weapons of Math Destruction by Cathy O'Neil, Crown Books, 2016

**Reference Books:**

1. Towards a code of ethics for artificial intelligence, Paula Boddington, Springer International Publishing, 2017
2. Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy, C. O'Neil, Broadway Books, 2016
3. The Black Box Society. Harvard University Press, F. Pasquale, 2015

Course Code	Course Name	L-T-P:C
HS105	History of Mathematics and Computing	2-0-0:2

**Objectives:** By the end of the course, students will:

1. Understand how mathematical ideas evolved across civilizations.

2. Appreciate India's foundational contributions such as zero, the decimal place-value system, and algebraic methods.
3. Analyze selected problems from historical mathematical texts.
4. Recognize the transmission of mathematical knowledge between cultures.
5. Understand the historical progression of computational devices.
6. Analyze the transition from mechanical calculation to digital computation.

**Pre-requisite:** None

**Contents:**

**Module I:** Mathematics in Ancient Civilizations (Egypt, Mesopotamia, and Greece), Mathematics in the Vedas and ritual practices: geometric constructions in Śulbasūtra, decimal place-value system and formal use of zero, Computational algorithms and arithmetic methods, Contributions of Aryabhata, Quadratic equations and indeterminate equations, Contributions of Brahmagupta, Contributions of Bhaskara II, Infinite series expansions for trigonometric functions, Iterative methods and approximations, Contributions of Madhava of Sangamagrama.

**Module II:** Spread of the Hindu-Arabic numeral system to the Islamic world and Europe, Revival of mathematics in Europe through translations, Solutions of cubic and quartic equations, development of analytic geometry, development of calculus by Newton and Leibniz.

**Module III:** Turing Machines, Cellular Automata, Finite State Machines, Graph Automata, Circuits, Tilings, Lambda Calculus, Fractran, Chemical Reaction Networks, Hopfield Networks, String Rewriting Systems, Tag Systems, Diophantine Equations, Register Machines, Primitive Recursive Functions, and more.

**Textbooks:**

1. History of Hindu Mathematics. A Source book. Parts 1 & 2. Bibhutibhusan Datta and Avadesh Narayan Singh, Asia Publishing House, 1935.
2. The History of Mathematics. An introduction. D. Burton, 7th ed, Mcgraw Hill, 2011.
3. A History of Modern Computing, Paul E. Ceruzzi, 2nd ed, MIT Press, 2003.
4. The Mathematics of India: Concepts, Methods, Connections. P. P. Divakaran. Hindustan Book Agency (India), 2018.
5. (Stillwell) Mathematics and its History, by John Stillwell; Springer (Undergraduate Texts in Mathematics).

Course Code	Course Name	L-T-P: C
IK101	Introduction to Yoga	0-2-0: 2

**Objectives:** The course is designed to provide students with a comprehensive understanding of the principles, practices, and benefits of yoga. The course aims to introduce students to the various aspects of yoga, including physical postures (asanas), breathing techniques (pranayama), meditation, and the philosophical foundation of yoga. Students will gain both theoretical and practical knowledge to incorporate yoga into their daily lives for physical, mental, and emotional well-being..

**Learning Outcomes:**

1. Understand the Basics of Yoga.
2. Learn Basic Yoga Postures (Asanas).
3. Develop Breathing Techniques (Pranayama).
4. Incorporate Yoga into Daily Life.

**Prerequisite:** None

**Contents:**

*Introduction to Yoga:* History and evolution of Yoga, fundamental principles of Yoga.

*Yoga Asanas:* Standing, seating, bending, and balance poses

*Pranayamas:* Importance of breath control during Yoga, connection between breath, body, and mind, Pranayamas techniques

*Meditation:* Role of meditation in Yoga, different forms of meditation, mindfulness meditation, Yoga for mental health, Yoga for emotional well-being.

Course Code	Course Name	L-T-P: C
NA101	NCC/NSS/NCA/NSO-I	0-0-0: 0

The National Service Scheme (NSS) at IIIT Vadodara aims to cultivate social responsibility, leadership, and civic engagement among students through structured volunteering activities. Drawing inspiration from the NSS chapters at leading institutions, this program outlines a balanced academic-year plan that integrates learning with impactful community service, aligned with national initiatives such as Swachh Bharat Abhiyan, Digital India, and environmental sustainability.

National Cultural Academy (NCA) is one of the programs, along with the above credited programs, that form a compulsory part of a student's academic record

at IIIT Vadodara. NCA provides an interesting and exciting opportunity to students to pursue their passions in a variety of cultural courses offered under its umbrella. Students are exposed to an experiential-based learning program with high-quality trainers to hone their creative skills while meeting curriculum requirements.

The National Sports Organisation (NSO) aims to promote physical fitness, teamwork, and sportsmanship among students as an integral part of their academic development. It provides opportunities to engage in various sports and fitness activities, encouraging a healthy and active lifestyle through experiential learning. NSO is a credit-based program within the curriculum. Students will participate in regular sports activities under trained coaches, with access to adequate infrastructure and facilities. Performance will be evaluated based on participation, discipline, and skill development, along with opportunities to compete at intra- and inter-institute levels.

The NCC unit aims to develop discipline, leadership, and a sense of patriotism among students. It prepares them to become responsible citizens through structured training and community service. The institute established an NCC unit in collaboration with the National Cadet Corps. The program will include drills, camps, and social service activities, with a credit-based evaluation system. Students will also have opportunities to earn NCC certifications and participate in national-level camps.

## Semester-II

### Courses and Contents

Course Code	Course Name	L-T-P: C
MA102	Mathematics-II (Discrete Mathematics)	3-1-0: 4

**Objective:** This course provides a foundation in discrete mathematics, covering logic, set theory, number theory, counting principles, recursion, relations, and graph theory. It develops logical reasoning and problem-solving skills with applications to algorithms, supported by practical exploration through selected topics from Thirty-three Miniatures by Jiří Matoušek.

**Learning Outcomes:** On successful completion of this course, students should be able to:

1. Apply principles of logic, set theory, and number theory to solve mathematical and computational problems.
2. Number Theory plays a fundamental role in building the foundation of cybersecurity and cryptography.
3. Formulate and solve recurrence relations and understand their role in algorithmic analysis.
4. Represent and analyze relations and graphs, and apply graph algorithms to real-world problems.

**Prerequisite:** None

#### Contents:

**Module I:** Language of Mathematics (Logic & Proof): Propositional logic, logical equivalences, predicates and quantifiers, translation from language to logical expressions, nested quantifiers, rules of inference, proof methods, direct proof, proof by contradiction, proof by contrapositive.

**Module II:** Language of Mathematics (Set Theory): Sets, axioms, set operations, set identities, function, inverse and composition of functions, cardinality of sets, countable and uncountable sets, Bernstein-Schröder Theorem, continuum hypothesis.

**Module III:** Relations: Relations and their properties, equivalence relations and equivalence classes, n-ary relations and their applications, representing relations, closures of relations, partial ordering, Hasse diagram, Lattice.

**Module IV:** Number Theory: Divisibility and modular arithmetic, integer representations, gcd and lcm, solving congruences, Chinese remainder theorem (CRT), Fermat's Theorem, Euler's Theorem, Diophantine equations and cryptography.

**Module V:** Counting: Permutation and combinations, pigeonhole principle, inclusion-exclusion principle, binomial theorem, Pascal identity and triangle.

**Module VI:** Induction & Recursion: Mathematical induction and second law of mathematical induction, recursive functions, recursive algorithm, recurrence relations, solving recurrence relations, generating functions.

**Module VII:** Graphs: Introduction and terminology, representation, isomorphism, connectivity, Warshall's algorithm, Euler and Hamilton path, shortest path problem, Dijkstra's algorithm, planar graph, Euler's formula, coloring of graphs.

#### Text Books:

1. Discrete Mathematics and its Application, 7th Ed, K. Rosen, Tata McGraw Hill, 2011.

#### Reference Books:

1. Discrete Mathematical Structure, 4th Ed, B. Kolman, R.C. Busby and S. C. Ross, PHI, 2000.
2. Combinatorics and Graph Theory, John M. Harris • Jeffry L. Hirst • Michael J. Mossinghoff.

Course Code	Course Name	L-T-P: C
MA104	Mathematics-II (Calculus)	3-1-0: 4

**Objective:** This course aims to build a solid foundation in calculus and differential equations, covering limits, continuity, differentiation, integration, and their applications. It also focuses on solving ordinary and partial differential equations using analytical methods and applying these concepts to real-world scientific and engineering problems.

**Learning Outcomes:** After successful completion of the course, a students will be able to

1. Incorporate the knowledge of calculus and its subsequent engineering applications.
2. Form and solve ordinary differential equations with engineering applications.

**Prerequisite:** None

#### Contents:

**Module I:** Calculus: The convergence of sequences and series, review of limits, continuity, differentiability, mean value theorems, Taylor series. theorems of integral calculus, evaluation of definite and improper integrals, partial derivatives, maxima and minima, multiple integrals, line, surface and volume integrals.

**Module II:** Complex Analysis: Complex numbers and elementary properties. Complex functions: limits, continuity and differentiation. Cauchy-Riemann equations. Analytic and harmonic functions. Anti-derivatives and path (contour) integrals. Cauchy-Goursat Theorem. Cauchy's integral formula.

**Module III:** Differential Equations: First order equations (linear and nonlinear), higher order linear differential equations, Cauchy and Euler equations, methods of solution using variation of parameters, Complementary function and particular integral, partial differential equations, variable separable method, initial and boundary value problems.

**Text Books:**

1. Hass, Joel R., Christopher E. Heil, and Maurice D. Weir. Thomas' Calculus. 14th ed. Pearson, 2018.
2. T.M. Apostol, Calculus, Volumes 1 and 2, 2nd ed., Wiley, 2007.

**Reference Books:**

1. Kreyszig, E., “Advanced Engineering Mathematics”, 10th Ed., Wiley India Pvt. Ltd, 2015.
2. Jain, R. K. and Iyenger, S. R. K., “Advanced Engineering Mathematics”, 5 th Ed., Narosa Publishing House, 2017.
3. Thomas, G. B., Hass, J., Heil, C. and Weir M. D., “Thomas’ Calculus”, 14th Ed., Pearson Education.

Course Code	Course Name	L-T-P: C
AI102	Optimization Techniques	3-0-0: 3

**Objectives:** This course introduces the principal algorithms for linear, network, discrete, nonlinear, dynamic optimization and optimal control. Emphasis is on methodology and the underlying mathematical structures. Topics include the simplex method, network flow methods, branch and bound and cutting plane methods for discrete optimization, optimality conditions for nonlinear optimization, interior point methods for convex optimization, Newton’s method, heuristic methods, and dynamic programming and optimal control methods.

**Learning Outcomes:** Understand and apply Linear Optimization methods, Master discrete and Nonlinear Optimization techniques, Solve Large-Scale and Robust Optimization problems

**Prerequisites:** None

**Contents:** Introduction to Optimization, Applications of linear optimization, Geometry of linear optimization,

Simplex method, Duality theory, Sensitivity analysis, Integer Programming, Branch and bound and cutting planes, Applications of discrete optimization, Lagrangian methods, Applications of nonlinear optimization, Optimality conditions and gradient methods, Line searches and Newton’s method, Conjugate gradient methods, Affine scaling algorithm, Interior point methods, Dynamic programming, Large scale optimization, Network flows, Heuristics and approximation algorithms.

**Text Books:**

1. Bertsimas, Dimitris, and John Tsitsiklis. Introduction to Linear Optimization. Belmont, MA: Athena Scientific, 1997. ISBN: 9781886529199.
2. S. S. Rao. Engineering optimization- Theory and practice, New Age international Pvt. Ltd.
3. An introduction to optimization by E. K. P. Chong and S. H. Zak (Wiley)

**Reference Books:**

1. David G. Luenberger and Yinyu Ye. Linear and Nonlinear Programming. Springer, 2008, Third Edition.
2. Jorge Nosedal and Stephan Wright. Numerical Optimization. Springer, 2006. Springer Series in Operations Research.
3. Dimitri Bertsekas. Convex Optimization Theory. Athena Scientific, 2009.

Course Code	Course Name	L-T-P: C
MC102	Calculus	3-1-0: 4

**Objective:** This course aims to build a solid foundation in calculus and differential equations, covering limits, continuity, differentiation, integration, and their applications. It also focuses on solving ordinary and partial differential equations using analytical methods and applying these concepts to real-world scientific and engineering problems.

**Learning Outcomes:** After successful completion of the course, a students will be able to

1. Incorporate the knowledge of calculus and its subsequent engineering applications.
2. Form and solve ordinary differential equations with engineering applications.

**Prerequisite:** None

**Contents:**

**Module I:** Calculus: The convergence of sequences and series, review of limits, continuity, differentiability, mean value theorems, Taylor series. theorems of

integral calculus, evaluation of definite and improper integrals, partial derivatives, maxima and minima, multiple integrals, line, surface and volume integrals.

**Module II:** Complex Analysis: Complex numbers and elementary properties. Complex functions: limits, continuity and differentiation. Cauchy-Riemann equations. Analytic and harmonic functions. Anti-derivatives and path (contour) integrals. Cauchy-Goursat Theorem. Cauchy's integral formula.

**Module III:** Differential Equations: First order equations (linear and nonlinear), higher order linear differential equations, Cauchy and Euler equations, methods of solution using variation of parameters, Complementary function and particular integral, partial differential equations, variable separable method, initial and boundary value problems.

#### Text Books:

1. Hass, Joel R., Christopher E. Heil, and Maurice D. Weir. Thomas' Calculus. 14th ed. Pearson, 2018.
2. T.M. Apostol, Calculus, Volumes 1 and 2, 2nd ed., Wiley, 2007.

#### Reference Books:

1. Kreyszig, E., "Advanced Engineering Mathematics", 10th Ed., Wiley India Pvt. Ltd, 2015.
2. Jain, R. K. and Iyenger, S. R. K., "Advanced Engineering Mathematics", 5 th Ed., Narosa Publishing House, 2017.
3. Thomas, G. B., Hass, J., Heil, C. and Weir M. D., "Thomas' Calculus", 14th Ed., Pearson Education.

Course Code	Course Name	L-T-P: C
PH102	Physics-II (Waves and Electromagnetics)	3-1-0: 4

**Objectives:** The objective of this course is to introduce the behavior of electromagnetic waves. This also provides an understanding of theories of electrostatics, magnetism and electrodynamics with their applications.

**Learning Outcomes:** After successful completion of this course, students will be able to:

1. Apply vector calculus to analyze simple electrostatic and magnetostatic fields.
2. Able to perform calculations involving various differential operators as well as line and surface integrals relating to Gauss and Stoke's theorems.
3. Apply the principles of Coulomb's Law and Gauss's law to electric fields in various coordinate systems.
4. Identify the electrostatic boundary-value problems by application of Poisson's and Laplace's equations.

5. Understand the depth of static and time-varying electromagnetic field as governed by Maxwell's equations.
6. Formulate and analyses problems involving conducting media with planar boundaries using uniform plane waves.

**Prerequisite:** None

#### Contents:

**Module I:** Mathematical Foundations: Vector Calculus, Gradient, Divergence and Curl. Line, Surface and Volume integrals. Gauss's divergence theorem and Stokes' theorem in Cartesian, Spherical polar and cylindrical polar coordinates, Continuity equation.

**Module II:** Review of Electrostatics: Electrostatics in Vacuum-Discrete and Distributed Charges, Electrostatic Force, Scalar & Vector Potentials, Electrostatic Energy, Poisson and Laplace equation and its applications; Electrostatics in Dielectric Medium-Electric Polarization; Electric Displacement Vector, Dielectric Susceptibility, Energy in Dielectric Medium.

**Module III:** Review of Magnetostatics: Magnetic Fields and Forces, Biot-Savart law and Ampere's law, Magnetic Vector Potential, Magnetization-Diamagnetism, Paramagnet-ism and Ferromagnetism, Ampere's Law in Magnetized Materials-Auxiliary Field H, Magnetic permeability and susceptibility.

**Module IV:** Review of Electrodynamics: Electromotive force, Time-varying fields, Faraday's' law of electromagnetic induction, Self and Mutual Inductance, Displacement Current, Maxwell's equations in Free Space & Inside Matter, Energy and Momentum in Electrodynamics.

**Module V:** Electromagnetic Waves: Wave equation, Propagation of Electromagnetic waves in Free Space and in Conducting Medium-Reflection and Refraction, Transmission and Dispersion.

#### Text Book:

1. *Introduction to Electrodynamics*, Griffiths. D. J, Prentice Hall, 2007.

#### Reference Books:

1. W. H. Hayt and J. A. Buck, *Engineering Electromagnetics*, Tata McGraw Hill Education Pvt. Ltd, 2006.
2. Purcell. E.M, *Electricity and Magnetism*, Berkley Physics Course, V2, Tata McGraw Hill, 2008.
3. Feynman. R.P, Leighton. R.B, Sands. M, *The Feynman Lectures on Physics*, Narosa Publishing House, Vol. II, 2008. Hill, 2008.
4. G. B. Arfken, H. J. Weber and F. E. Harris, *Mathematical Methods for Physicists*, Academic Press, 2013.

Course Code	Course Name	L-T-P: C
PH162	Experimental Physics-II	0-0-2: 1

**Objectives:** The laboratory course provides hands-on experience to students with the basic experiments of electromagnetics and wave propagation.

**Learning Outcomes:** After successful completion of this course, students will be able to:

1. Visualize the nature of electric field lines under different physical conditions.
2. Distinguish electric and magnetic fields.
3. Measure magnitude of electronic charge and dielectric constant of a material.
4. Understand the effects of magnetic fields, Hysteresis effect.
5. Understand the principle of communications through propagation of electromagnetic waves in a different medium.

**List of Experiments:**

1. Measurement of elementary charge using Millikan oil drop experiment.
2. To draw electric field lines and equipotential lines.
3. Measurement of Dielectric constant of dielectric materials.
4. Measure Magnetic field of a paired coils in a Helmholtz arrangement with a Teslameter.
5. Verify Faraday Law and Induced *e.m.f.*
6. Hysteresis in Ferromagnetic Materials.
7. Microwave optic system to study properties of electromagnetic waves.
8. Experiments on Solar-Cell Trainer kit.
9. Dispersion of light in Prism Spectrometer.
10. Measurement of Permeability and Permittivity of Air and determination of speed of Light.

**Reference:** Laboratory Manual

Course Code	Course Name	L-T-P: C
PH104	Physics-III (Classical Physics)	3-0-0: 3

**Objectives:** This course provides a rigorous foundation in classical mechanics through Newtonian, Lagrangian, and Hamiltonian formalisms. Students will learn to analyze particle and rigid-body dynamics, apply symmetry and conservation laws, and solve constrained multi-degree-of-freedom systems. The course also introduces nonlinear and chaotic dynamics to connect classical theory with modern physical and engineering applications. By course end, students will be able to model real-world systems using advanced analytical tools.

**Learning Outcomes:** Upon successful completion of the course, students will be able to

1. Formulate and solve mechanical systems using Newtonian, Lagrangian, and Hamiltonian approaches, including constrained, multi-particle, and rigid-body problems.
2. Analyze nonlinear dynamics and interpret real-world physical systems through phase-space methods, symmetry principles, and numerical modeling.

**Prerequisites:** Undergraduate-level calculus, differential equations, and vector algebra, along with introductory physics covering Newtonian mechanics.

**Contents:**

**Module-I:** Momentum and angular momentum conservation; scattering; central force motion; Kepler problem; effective potential; two-body reduction

**Module-II:** Calculus of variations; Hamilton's principle; generalized coordinates; constraints; Euler-Lagrange equations; conservation laws & symmetries; Noether theorem (intro); small oscillations; normal modes

**Module-III:** Legendre transform; Hamilton's equations; canonical transformations; Poisson brackets; phase space & Liouville theorem; action-angle variables; Hamilton-Jacobi theory

**Module-IV:** Inertia tensor; Euler angles; Euler equations; rigid body rotation; symmetric and asymmetric top; gyroscopes; motion in rotating frames (Coriolis, centrifugal forces)

**Module-V:** Nonlinear oscillations; perturbation theory; resonance; stability analysis; phase portraits; bifurcations; chaos in Hamiltonian systems; Lyapunov exponents; Poincaré sections; optional introduction to classical fields

**Text Books:**

1. Herbert Goldstein, Charles P. Poole Jr., and John L. Safko, Classical Mechanics, 3rd Edition, Pearson Education (2002).

**Reference Books:**

1. L. D. Landau and E. M. Lifshitz, Mechanics, 3rd Edition, Course of Theoretical Physics, Vol. 1, Butterworth-Heinemann (1976).
2. John R. Taylor, Classical Mechanics, University Science Books (2005).
3. Stephen T. Thornton and Jerry B. Marion, Classical Dynamics of Particles and Systems, 5th Edition, Brooks/Cole (2003).
4. Stephen H. Strogatz, Nonlinear Dynamics and Chaos: With Applications to Physics, Biology,

Chemistry, and Engineering, 2nd Edition, Westview Press (2018).

5. A. I. Lichtenberg and M. A. Lieberman, Regular and Chaotic Dynamics, 2nd Edition, Springer-Verlag (1992).

Course Code	Course Name	L-T-P: C
CS102	Introduction to Data Structure	3-0-0: 3

**Objectives:** The course covers fundamental data structures, including arrays, linked lists, stacks, queues, heaps, binary trees, and graphs. Students will learn to implement generic data structures with key operations like insertion, deletion, and searching. Additionally, the course emphasizes algorithm design and analysis.

**Learning Outcomes:** On successful completion of this course, students should be able to:

1. Design and implement various data structures.
2. Analyze the strength and weakness of different data structures.
3. Compare various sorting and searching algorithms.
4. Design optimized solutions for computational problems using appropriate data structures.
5. Implement data structures in real-world applications and enhance problem-solving skills.

**Prerequisite:** CS101 and CS161

**Contents:**

**Module I:** Introduction- Representation of data on a computer, data types, abstract datatypes & array and linked list representations ways of representing programs and associated data on computers.

**Module II:** Analysis Tools- Notion of the running time of an algorithm, recurrences, parameters of performance.

**Module III:** Dictionary Operations- Find, max, min, successor, predecessor (query operations); insert, delete (modify operations) LIST DATA: Stacks, queues, variants implementation using arrays and linked lists.

**Module IV:** Sorting- Comparison based sorting algorithms, other sorting algorithms, lower bounds for comparison-based sorting algorithms best-case, worst-case and average-case running times; quicksort, heap Sort, insertion sort, bubble sort, etc.

**Module V:** Trees- Heaps, Binary search trees (BST), heights of BST BALANCED BSTs: Red Black trees, AVL Trees.

**Module VI:** Graphs- Representation using adjacency matrices and adjacency lists, Graph searching algorithms BFS and DFS.

**Text Books:**

1. Data Structures, Algorithms and Applications in C++, Sartaj Sahni, Universities Press, 2nd Edition (2009).
2. Data Structures and Algorithm in JAVA, Robert Lafre, Pearsons, 2nd Edition (2002).

**Reference Books:**

1. Introduction to Algorithms, 3th Ed, Cormen, Lieserson and Rivest, PHI, 2011.
2. Data Structures and Algorithms, Aho, Hopcroft and Ullman, Addison-Wesley, 1999.

Course Code	Course Name	L-T-P: C
CS162	Introduction to Data Structure Lab	0-1-2: 2

Lab and take home assignments based on the course “Data Structures”. Possible set of assignments may include the following list of experiments.

**List of Experiments:**

1. Design and Implement List data structure using i) array ii) singly linked list.
2. Design and Implement basic operations on doubly linked list.
3. Design and Implement stack using i) array ii) singly linked list.
4. Design and Implement Queue using i) array ii) singly linked list.
5. Design and Implement basic operations on Circular Queue.
6. Implementation of Searching algorithms (Linear search, Binary search).
7. Implementation of various sorting algorithms (Insertion sort, Bubble sort, Quick sort, Heap sort, Merge sort, counting sort, radix sort etc.).
8. Design and Implement basic operations (insertion, deletion, search, findmin and find max) on Binary Search trees.
9. Design and Implement basic operations of various balanced BSTs, e.g., AVL trees.
10. Implementation of Breadth First Search Techniques.
11. Implementation of Depth First Search Techniques.

Course Code	Course Name	L-T-P: C
-------------	-------------	----------

AI104	Data Visualization and Analytics	0-1-2:2
-------	----------------------------------	---------

**Objective:** Data analytics is a method of collecting both qualitative and quantitative information about properties of interest therefore allowing users to make informed decisions. The Visualization Techniques are an important tool in displaying such information in visual understanding of the given Data. In this course, students will be able to learn data analytic techniques to analyze datasets and make statistically valid conclusions and their respective visual representation. Students will also be exposed to a statistical software package for data analysis and Visualization.

**Learning Outcomes:** On successful completion of this course, students should be able to:

1. Master Data Visualization and Statistical Analysis.
2. Apply Advanced Data Analytics Technique for feature engineering and dimensionality reduction.
3. Analyze and Forecast Time Series Data.

**Prerequisites:** None

**Contents:**

**Module I:** Introduction to R and Tableau

**Module II:** Visualization Tools: Basic plotting structure and information: Pie chart, Bar Graphs, Histogram, Box Plot, Tree plot, Dendrogram, Scatter, Heat map.

**Module III:** Hypothesis Testing: Introduction to Hypothesis Testing, P- value and h- value analysis, Confidence Interval, Type -I and Type-II errors, chi-square test, t-test, t-SNE, Wilcoxon test, ANOVA, R-square test.

**Module IV:** Data Analytics Techniques: Feature Engineering – Feature selection, Normalization, Feature Reduction, Wrapper method, Feature Importance and Ranking. Dimensionality Reduction Techniques – PCA, Linear Discriminant Analysis, Locally Preserving Projections

**Module V:** Time Series Analysis: Stochastic Process, Moving Average Method, Regression in Time Series data, ARIMA Model,

**Text Books:**

1. John Maindonald and W. John Braun, Data Analysis and Graphics using R, 3rd Ed., Cambridge University Press, 2010.
2. Michael Baron, Probability and Statistics for Computer Scientists, 3rd Ed, CRC Press, 2019

**Reference Books:**

1. Robert Shaumway and David S. Stoffer., Time Series Analysis and its Applications (with R Examples), 4th Ed, Springer 2016.
2. Ronald Walpole, Raymond Myers, Sharon Myers and keying Ye, Probability & Statistics for Engineers and Scientists, 9th Ed. , Pearson 2019

Course Code	Course Name	L-T-P: C
EC102	Basic Electronics-II	3-1-0: 4

**Objectives:** The course objective is to provide students with a comprehensive understanding of analog electronic circuits, focusing on their analysis, design, and practical applications. The course objectives include: Gaining knowledge about the characteristics and functions of various electronic components such as diodes, transistors, and operational amplifiers. Developing the ability to analyze and design analog circuits, including amplifiers, oscillators. Applying theoretical concepts to hands-on experience with circuit construction, SPICE simulations and testing.

**Learning Outcomes:** On successful completion of this course, students should be able to:

1. Developing a deep understanding of analog circuit behavior and the ability to design and analyze electronic systems.
2. Design and analyze fundamental analog circuits such as amplifiers, oscillators, and filters using various active and passive components.
3. Learn to model and use semiconductor devices like diodes and transistors for practical circuit applications.
4. Develop the ability to troubleshoot analog circuits, identify issues, and optimize their performance by adjusting circuit parameters.

**Prerequisite:** Fundamentals of semiconductor devices and circuits.

**Contents:**

**Module-I:** BJT, FET Amplifiers and Small Signal Model: Load line concept, operating points, Transistor Bias Circuits, Voltage Amplifiers, Common Collector and Common Base Amplifiers, Common Drain, Common Gate, Common Source and Frequency Response, Two port device and Hybrid model, Miller's Theorem, Differential amplifier.

**Module-II:** Multistage Amplifiers: Multistage amplifier, coupling mechanism, effect of coupling and bypass capacitors, Design of single stage and multistage RC coupled amplifier, Analysis of Cascaded RC Coupled amplifiers, Cascode amplifier, Darlington pair,

Current mirrors, High frequency pi Model, High frequency analysis.

**Module-III:** Operational Amplifier: Differential Amplifiers, Op-Amp pin configuration, Ideal and Practical Characteristics of Op-Amp, Inverting and Non-Inverting Amplifiers, Active Filters, Summing Amplifiers, Differential and Integrating Amplifiers, Comparators, Frequency response of an Op-Amp.

**Module-IV:** Feedback Amplifiers: Introduction, types of amplifier, concept of feedback, feedback topology, effect on gain, bandwidth, input and output resistance, Use of positive feedback.

**Module-V:** Oscillators: Barkhausen criterion for oscillations, sinusoidal oscillators, phase shift, resonant circuit oscillators, crystal oscillator. Different oscillator circuits-tuned collector, Hartley Colpitts, phase shift, Wien's bridge, and crystal oscillator, 555 Timer and Circuits.

**Text Books:**

1. Microelectronic Circuits: Theory and Applications, A.S. Sedra and K.C. Smith, Oxford University Press, 7th Edition.
2. Integrated Electronics, Millman, Jacob. Millman's, 2E. Tata McGraw-Hill education, 2010.
3. Electronic Principles, 7th Ed, Albert Malvino, and David Bates, Tata McGraw-Hill, 2007.

**Reference Books:**

1. Electronic Devices; 9th Edition, Thomas L. Floyd, Pearson.
2. Introduction to PSpice using OrCAD for circuits and electronics, M.H. Rashid, 2003. Prentice-Hall, Inc., 2003.
3. Electronic devices and circuit theory, R.L. Boylestad and L. Nashelsky, Pearson Educación, 2002.

Course Code	Course Name	L-T-P: C
EC162	Basic Electronics Lab-II	0-0-3: 2

**Objectives:** The course objective is to provide students with a comprehensive understanding of analog electronic circuits, focusing on their analysis, design, and practical applications. The course objectives include: Gaining knowledge about the characteristics and functions of various electronic components such as diodes, transistors, and operational amplifiers. Developing the ability to analyze and design analog circuits, including amplifiers, oscillators. Applying theoretical concepts to hands-on experience with circuit construction, SPICE simulations and testing.

**Prerequisite:** None

**Laboratory Outcomes:**

1. Understand the analog circuit and its design using SPICE.
2. Apply a small signal amplifier model for different circuits.
3. Analyze cascaded amplifier configurations to obtain the required overall specifications.
4. Design different amplifiers.
5. Build analog subsystems using amplifiers and oscillators.

**List of Experiments:**

1. To study the basics of the SPICE model and defining the passive and active elements.
2. To Study the RLC circuits using the SPICE tool.
3. To study different operating points and biasing circuits of BJT.
4. To study common emitter amplifiers and Common Source amplifiers.
5. To design a two stage RC coupled common emitter transistor (n-p-n) amplifier circuit and determine its frequency response curve.
6. To design a Common-Emitter transistor (n-p-n) amplifier circuit, obtain the frequency response curve of the amplifier and determine the mid-frequency gain, Amid, lower and higher cut-off frequencies.
7. To design Inverting and Non-Inverting Operational Amplifiers for a given specification.
8. To design Integrator and Differentiator using Operational Amplifier.
9. To design an active low pass and high pass filter of different orders using Op-Amplifier.
10. To obtain frequency response of single stage CE amplifiers.
11. To study the effect of negative feedback with and without bypass capacitors.
12. To obtain frequency response of two stage RC coupled amplifiers.
13. To study feedback amplifiers.
14. To study RC phase shift oscillators.

**Reference:** Laboratory Manual

Course Code	Course Name	L-T-P: C
CC102	Python Programming	1-0-3: 3

**Course Objectives:** To introduce the fundamentals of Python programming and core python data structures for developing problem-solving and algorithmic

thinking skills to solve simple cyber security related problems.

**Learning Outcomes:** Upon successful completion of this course, students will be able to:

1. Develop and debug basic python program using proper syntax, control structures and I/O operations in configured python environments.
2. Implement and manipulate fundamental data structures such as Lists, Tuples using indexing, slicing, methods and number conversion.
3. Design modular Python applications using various functions with parameters, expressions, namespaces, and code organization techniques for software solutions.
4. Apply OOP principles and utilize different libraries for data manipulation, preprocessing, exploration, and plotting.
5. Handle exceptions/file operations securely, and develop applications for network scanning, encryption, log parsing, and threat detection using various libraries.

**Prerequisites:** CS101

**Contents:**

**Module I:** Introduction to Python, setting up the python environment (installation, IDEs), Programming concepts: variables, data types, operators, expressions, boolean logic, Basic syntax and program structure (input/output, control flow statements: IF-Else, Loops: For, While) , Debugging techniques.

**Module II:** Data Structures: Lists, tuples, dictionaries, sets (operations, methods and applications), String, String manipulation (subscript, operator, indexing, slicing, string methods), strings and number system: converting strings to numbers and vice versa. Binary, octal, hexadecimal numbers.

**Module III:** Defining and calling functions, arguments-formal and actual arguments, named arguments, parameters, and return values , Modular programming concepts (importing modules, creating modules), namespaces and scope, Lambda function, Recursive function. Scripts: splitting codes.

**Module IV:** Introduction to OOP concepts (classes, objects, attributes, methods), Creation and use of classes, Inheritance and Polymorphism, Encapsulation and Data abstraction, Introduction to NumPy and Pandas libraries, Data manipulation and exploration, Data preprocessing and plotting tools (Matplotlib).

**Module V:** Exceptions and error handling mechanisms, Reading and writing data from files, File operations and error handling, Applications: network scanning,

encryption, log parsing, and threat detection using libraries like Scapy, Cryptography, and Pandas.

**Textbooks:**

1. Fundamentals of Python: First Programs by Kenneth A. Lambert, Cengage Learning, Inc., 3rd edition, 2023.
2. Python for Everybody: Exploring Data Using Python 3 by Charles R. Severance, Shroff Publishers, 3rd edition, 2016.

**Reference Books:**

1. Python Programming using Problem Solving Approach by Reema Thareja, Oxford University Press, 2nd Edition, 2023
2. Think Python: How to Think Like a Computer Scientist by Allen B. Downey, O’reilly, 2nd edition, Updated for Python 3, Shroff/O’Reilly, 2016.
3. Introduction to Computation and Programming Using Python by John V Guttag MIT Press, Revised and expanded Edition, 2013.

Course Code	Course Name	L-T-P: C
CC104	Modern Cryptography	3-1-0: 4

**Objectives:** To provide fundamental understanding of modern cryptographic techniques and their application in securing digital communication and information systems.

**Learning Outcomes:** Upon successful completion of this course, students will be able to:

1. Understand basic concepts of cryptography and security goals.
2. Apply symmetric and asymmetric encryption techniques.
3. Use hashing and digital signatures for data integrity and authentication.
4. Analyse cryptographic protocols and their security.

**Prerequisites:** CC101

**Contents:**

**Module I:** Introduction and Classical Cryptography. The Setting of Private-Key. Historical Ciphers. Principles of Modern Cryptography. Provable Security and Real-World Security. Perfectly Secret Encryption. Definitions. The One-Time Pad. Limitations of Perfect Secrecy. Shannon’s Theorem.

**Module II:** Symmetric Cryptography. Private-Key Encryption. Computational Security. Defining Computationally Secure Encryption. Constructing Secure Encryption Schemes. Stronger Security Notions. Constructing CPA-Secure Encryption Schemes. Modes of Operation. Chosen-Ciphertext Attacks. Message Authentication Codes. Message Integrity. Message Authentication Codes. Constructing Secure Message Authentication Codes. CBC-MAC. Authenticated Encryption. Information-Theoretic MACs.

**Module III:** Hash Functions and Applications. Definitions. Domain Extension. Message Authentication Using Hash Functions. Generic Attacks on Hash Functions. The Random-Oracle Model. Additional Applications of Hash Functions: Fingerprinting and Deduplication, Merkle Trees, Password Hashing, Key Derivation, Commitment Schemes.

**Module IV:** Practical and Theoretical Constructions of Symmetric-Key Primitives. LFSR based stream ciphers. RC4. DES. AES. SHA3. One-Way Functions. From One-Way Functions to Pseudorandomness. Hard-Core Predicates from One-Way Functions. Constructing Pseudorandom Generators. Pseudorandom Generators with Minimal Expansion. Constructing Pseudorandom Functions. Constructing (Strong) Pseudorandom Permutations. Assumptions for Private-Key Cryptography. Computational Indistinguishability.

**Module V:** Asymmetric Cryptography. Number Theory and Cryptographic Hardness Assumptions. Preliminaries and Basic Group Theory. Primes, Factoring, and RSA. Cryptographic Assumptions in Cyclic Groups. Cryptographic Applications. Algorithms for Factoring and Computing Discrete Logarithms. Key Management. Public-Key Encryptions and their security analysis. Digital Signature Schemes and their security analysis. Public-Key Encryption from Trapdoor Permutations. The Paillier Encryption Scheme. Secret Sharing and Threshold Encryption. Interactive Proofs and Zero Knowledge.

#### Text Books:

1. Introduction to Modern Cryptography by Jonathan Katz, Yehuda Lindell.
2. Lecture Notes on Cryptography by Shafi Goldwasser and Mihir Bellare

#### Reference Books:

1. Handbook of Applied Cryptography by Scott Vanstone, Alfred Menezes, Paul van Oorschot.
2. Cryptography Theory and Practice by Douglas Stinson.

### 3. Foundations of cryptography by Oded Goldreich

Course Code	Course Name	L-T-P: C
HS102	Science, Technology and Society	3-0-0: 3

**Objectives:** The course aims to provide a grounding in the historical and philosophical context of science, technology and society (STS) in India. It aims to develop students' capacities for social awareness and for critical thinking, and establish an understanding of the historical and contemporary interrelationship between science, technology and society as they evolve together.

**Learning Outcomes:** On successful completion of this course, students should be able to:

1. Apply knowledge of core theories, methods and its applications to contemporary society in the interdisciplinary field of STS.
2. Apply social/scientific theories and methods to analyze interactions between science, technology and society in particular historical, political and cultural contexts.
3. Evaluate critically the empirical evidence and theoretical claims in STS-related debates.
4. Communicate clearly and persuasively about STS issues to a general audience using different forms of communication including oral presentation and writing.
5. Compare and contrast Indian philosophical approaches to knowledge (e.g., Nyaya, Samkhya) with modern scientific methods.
6. Apply concepts from IKS to contemporary issues such as sustainability, health, and technology and propose ways to integrate traditional knowledge with modern scientific research or innovation.

#### Contents:

**Module I:** Introduction to STS as a field of study and research in the twentieth century:

1. STS as Quest for Knowledge.
2. Nature of Science and Distinct Characteristics of Science.
3. Science in Twentieth Century.
4. Science and Technology in Societal Context
5. Interdisciplinary nature: sociology, history, philosophy of science.
6. Contributions of Karl Popper (Falsifiability as criterion of science), Robert K. Merton-Norms of science (CUDOS: Communalism, Universalism, Disinterestedness, Organized Skepticism).
7. Major transformations: Rise of "Big Science" (e.g., nuclear research, space programs),

Institutionalization of research and funding, Science during wars (e.g., atomic research), Paradigm shifts and scientific revolutions, Thomas Kuhn- Concept of paradigms and scientific revolutions.

8. Role of culture, politics, and economics in shaping science, Critiques of scientific neutrality, Knowledge-power relations.
9. Understanding cyber-culture: digital society, virtual identities, networks

**Module II:** Philosophical, Historical and Sociological Approaches to STS:

1. Philosophy of Science and its Interconnection with Technology and Society.
2. Scientific explanation, objectivity, and truth.
3. Science and Western Civilization.
4. Social construction of scientific knowledge -Actor-Network Theory (Bruno Latour).
5. Social Theorists.
6. Relationship between Literature, Society and Science.
7. Scientific/Technological and Sociological Literacy.
8. Science in the social world: institutions, actors, and practices.
9. Demarcation problem: science vs pseudoscience.
10. Case discussions: Marxism (associated with Karl Marx- debate on whether it is scientific or ideological), Freudian Psychoanalysis (by Sigmund Freud-critique as unfalsifiable theory).
11. Science and Technology a Social Institution.
12. Technoscience and Globalizations.
13. Life after STS

**Module III:** The growth and identity of Modern Science and Technology in India:

1. History and key themes.
2. Modernism and Postmodernism.
3. Science, Policy and Nation-Building in India.
4. Pioneers of Modern Indian Science.
5. Architects of India's Scientific Institutions.
6. Indian Scientists in Applied and Agricultural Sciences.
7. Scientists as Policy Makers and Advisors.
8. Contributions to Strategic and Public Policy.
9. Contemporary Science Policy and Public Health Experts.
10. Science, Innovation and Self-Reliance (Post-Independence Vision).
11. Impact on Indian Society and Global Science.
12. Pace of Innovation.
13. Privileged Positions of Business and Science.

14. STS as Social Construction

**Module IV:** Science Communication: Institutions, Ideologies, Practices

1. The diversity of science communication in colonial India.
2. Science communication and Policies in India: Historical Perspective.
3. Modes and Means of Science communication in India.
4. Role of Technical Education and Engineering Profession.
5. Wealth, Engineering Institutions and Engineering Education.
6. Role of Public Funded Institutions, and Challenges.
7. Role of Various Agencies in Technical Education.
8. Educational Technologies and Pedagogy.
9. Computers. Learning, and Constructivist Theory.
10. Affordances for Learning and Teaching and Online Learning Environments

**Module V:** Science Communication and the Postcolonial Thought:

1. Nature of Nehruvian Science and Gandhian Thought.
2. Concept of Scientific Temper.
3. Decolonization of science and knowledge production.
4. Science as a tool for nation-building and industrial growth.
5. Establishment of scientific institutions (IITs, CSIR labs).
6. Postcolonial critique of Western dominance in science.
7. STS and Postcolonialism.
8. Scientific Policy Resolution.
9. Problems Plaguing Practice of Science in India.
10. Role of government in funding and guiding research.
11. Link between science and economic development.
12. Expansion of research institutions and infrastructure

**Module VI:** Liberalization and the commoditization of science and technology:

1. Impact of Science on Society and Globalization.
2. Economic Liberalization in India (post-1991 reforms).
3. Science and technology as market-driven commodities.

4. Role of private sector and multinational companies (e.g., Infosys, Tata Consultancy Services).
5. Intellectual Property Rights and patents (World Trade Organization framework).
6. Commercialization of research and innovation.
7. Rise of start-ups and tech entrepreneurship.
8. Inequality in access to scientific advancements.
9. Improvements in healthcare, communication, and transportation.
10. Role of Digital Revolution.
11. Privatization and Learning Science outside the Classroom (e.g., BYJU'S).
12. Science learning through exhibitions, experiential learning via labs, workshops, and fieldwork.
13. Impact of privatization on accessibility and quality of education.
14. Inequality between public and private education systems.
15. ICT as Secondary Science.
16. Innovation in Instructional Technology.
17. Use of Artificial Intelligence in education (adaptive learning systems- AR, VR, LLM, Moodle etc).
18. Ethics and Cybercrime.
19. Types of Cybercrime (hacking, phishing, identity theft).
20. Data privacy and protection issues.
21. Role of laws and regulations (e.g., Information Technology Act 2000).
22. Ethical concerns in AI and data usage

**Module VII:** Evolution of Scientific Thought in 21st India:

1. Science as a socio-cultural construct and restructuring of Indian knowledge.
2. Emergence of plural knowledge systems in contemporary discourse.
3. Definition, scope, and classification of Indian Knowledge Systems (IKS).
4. Policy Perspective: NEP 2020 and IKS Integration.
5. Indian Philosophical Foundations of Scientific Thought: Concepts such as *Pramana* (means of knowledge), *Anumana* (inference), *Pratyaksha* (perception); Schools of thought: Nyaya (logic and reasoning), Vaisheshika (atomism and categories of reality), Samkhya (cosmology and causation).
6. Holistic vs reductionist approaches to science.
7. Science, Sustainability, and Indigenous Practices.
8. Ecological wisdom in traditional agriculture and water systems.
9. Climate resilience and local knowledge.

10. Ethics of coexistence and environmental balance.

**Text Books:**

1. Arnold, D. (2004). *Science, technology and medicine in colonial India*. Cambridge University Press.
2. Chakrabarti, P. (2004). *Western science in modern India: Metropolitan methods, colonial practices*. Permanent Black.
3. Bell, D. (2001). *Cyberculture: The key concepts*. Routledge.
4. Bell, D. (2007). *Cyberculture theorists: Manuel Castells and Donna Haraway*. Routledge.
5. Bauchspies, W. K., Croissant, J., & Restivo, S. (2005). *Science, technology, and society: A sociological approach*. Blackwell Publishing.
6. Rosenberg, A. (2012). *Philosophy of science: A contemporary introduction* (3rd ed.). Routledge.
7. Government of India. (2020). *National education policy 2020*. Ministry of Human Resource Development.
8. Kapoor, K. (2005). *Text and interpretation: The Indian tradition*. DK Printworld.
9. Dharampal. (1971). *Indian science and technology in the eighteenth century: Some contemporary European accounts*. Impex India.
10. Chattopadhyaya, D. (1977). *Science and society in ancient India*. Research India Publications.

**Reference Books:**

1. Bose, D. M., Sen, S. N., & Subbarayappa, B. V. (2009). *A concise history of science in India*. University Press.
2. Sherman, W. R. (2000). *Science and society in the twentieth century*. Routledge.
3. McComas, W. F. (Ed.). (1998). *The nature of science in science education: Rationales and strategies*. Kluwer Academic Publishers

Course Code	Course Name	L-T-P: C
HS104	Disaster Management	2-0-0: 2

**Description:** This course provides an interdisciplinary understanding of disaster management. It introduces students to different types of disasters such as floods, earthquakes, cyclones, and wildfires, along with their causes, impacts, and mitigation strategies. This course highlights the importance of sustainable development, climate adaptation, and disaster risk reduction (DRR) frameworks. Through case studies, practical examples, and project-based learning, students will gain the skills needed to develop innovative and technology-driven solutions for real-world disaster management

challenges. It will be equipped to the students with integrated technical knowledge with disaster management principles and policies.

#### Objectives:

1. To develop a strong foundation in natural hazards and disaster management principles.
2. To enable students to understand risks, impacts, and mitigation strategies.
3. To Promote knowledge of disaster-resilient and sustainable development practices, aligning with global goals like disaster risk reduction and climate adaptation.
4. To encourage knowledge dissemination and innovation, enabling students and creating awareness to learn about tools, and community-based technological solutions for effective disaster management.

#### Course Outcomes:

1. Gain comprehensive knowledge of natural hazards and disaster management concepts, including risk, vulnerability, and resilience.
2. Promotes interdisciplinary thinking by integrating technology with environmental and social challenges.
3. Develops a basic understanding of natural and technical hazards. Enhances knowledge of disaster management concepts, including risk assessment, preparedness, mitigation, and response.

#### Contents:

##### Module I: Understanding Disaster Management

Understanding of key concepts in Disaster Management: Hazards, Disasters, Vulnerability, Resilience, Disaster Management, Disaster Cycle, risk, prevention, mitigation, relief and response, recovery and rehabilitation etc – Brief history of disaster management in India and world. Role of social sciences and natural sciences and multidisciplinary nature of disaster management as applied disciplines

##### Module II: Types of Disaster and Case Studies

Different Types of Disaster: Consequences and impact of disasters; tsunamis, earthquakes, landslides, cyclones, Wildfire, volcanic eruption- Overview through case studies

##### Module III: Study of Important disasters

Geological disaster and its types, magnitude and intensity, seismic zones of India, major fault systems of India plate, flood types and its management, drought types and its management, Technical disaster (Chemical, Biological and Nuclear), Overview through case studies. And its management, Social Economics and Environmental impact of disasters.

##### Module IV: Mitigation Strategies and Disaster Management Policy

Mitigation and Management techniques of Disaster Basic principles of disaster management, Disaster Management cycle, Disaster management policy, National and State Bodies for Disaster Management, Building design and construction in highly seismic zones, retrofitting of buildings, Early Warning Systems, Building design and construction in highly seismic zones, retrofitting of buildings

##### Module V: Awareness Program and Project on Disaster Management

Awareness generation program for disaster preparedness, Usages of GIS and Remote sensing techniques in disaster management, Mini project on disaster risk assessment and preparedness for disasters

#### Text Books:

1. Dr. Mrinalini Pandey, Disaster Management Wiley India Pvt. Ltd.
2. Tushar Bhattacharya Disaster Science and Management McGraw Hill Education (India) Pvt. Ltd.
3. Jagbir Singh Disaster Management : Future Challenges and Opportunities K W Publishers Pvt. Ltd.
4. Gupta A.K., Niar S.S and Chatterjee S. (2013) Disaster management and Risk Reduction, Role of Environmental Knowledge, Narosa Publishing House, Delhi.
5. World Disasters Report, 2009. International Federation of Red Cross and Red Crescent, Switzerland.
6. NIDM Publications: Publications of National Disaster Management Authority (NDMA) on Various Templates and Guidelines for Disaster Management.

Course Code	Course Name	L-T-P: C
NA102	NCC/NSS/NCA/NSO-I	0-0-0: 0

The National Service Scheme (NSS) at IIIT Vadodara aims to cultivate social responsibility, leadership, and civic engagement among students through structured volunteering activities. Drawing inspiration from the NSS chapters at leading institutions, this program outlines a balanced academic-year plan that integrates learning with impactful community service, aligned with national initiatives such as Swachh Bharat Abhiyan, Digital India, and environmental sustainability.

National Cultural Academy (NCA) is one of the programs, along with the above credited programs, that form a compulsory part of a student's academic record at IIT Vadodara. NCA provides an interesting and exciting opportunity to students to pursue their passions in a variety of cultural courses offered under its umbrella. Students are exposed to an experiential-based learning program with high-quality trainers to hone their creative skills while meeting curriculum requirements.

The National Sports Organisation (NSO) aims to promote physical fitness, teamwork, and sportsmanship among students as an integral part of their academic development. It provides opportunities to engage in various sports and fitness activities, encouraging a healthy and active lifestyle through experiential learning. NSO is a credit-based program within the curriculum. Students will participate in regular sports activities under trained coaches, with access to adequate infrastructure and facilities. Performance will be evaluated based on participation, discipline, and skill development, along with opportunities to compete at intra- and inter-institute levels.

The NCC unit aims to develop discipline, leadership, and a sense of patriotism among students. It prepares them to become responsible citizens through structured training and community service. The institute established an NCC unit in collaboration with the National Cadet Corps. The program will include drills, camps, and social service activities, with a credit-based evaluation system. Students will also have opportunities to earn NCC certifications and participate in national-level camps.

---

## Semester-III

### Courses and Contents

Course Code	Course Name	L-T-P: C
SC201	Environmental Science	2-0-0: 2

**Objectives:** The course on Environment Science is expected to give information about the environment that will lead to a concern for environment. When one develops this concern, he/she will begin to act at his/her own level to protect the environment.

**Learning Outcomes:** On successful completion of this course, students should be able to:

1. Understand environmental problems arising due to developmental activities.
2. Identify the natural resources and suitable methods for conservation and sustainable development.
3. Realize the importance of ecosystem and biodiversity for maintaining ecological balance.
4. Identify the environmental pollutants and abatement devices.

**Prerequisite:** None

**Contents:**

**Multidisciplinary nature of environmental studies:** Definition, Scope and Importance, Need for public awareness.

**Natural Resources:** Renewable and non-renewable resources; Natural resources and associated problems:

1. Forest resources: Use and over-exploitation, deforestation, case studies. Timber extraction, mining, dams and their effects on forest and tribal people.
2. Water resources: Use and over-utilization of surface and ground water, floods, drought, conflicts over water, dams-benefits and problems.
3. Mineral resources: Use and exploitation, environmental effects of extracting and using mineral resources, case studies.
4. Food resources: World food problems, changes caused by agriculture and overgrazing, effects of modern agriculture, fertilizer-pesticide problems, water logging, salinity, case studies.
5. Energy resources: Growing energy needs, renewable and non-renewable energy sources, use of alternate energy sources. Case studies.
6. Land resources: Land as a resource, land degradation, man induced landslides, soil erosion and desertification.

Role of an individual in conservation of natural resources. Equitable use of resources for sustainable lifestyles.

**Ecosystems:** Concept of an ecosystem, Structure and function of an ecosystem, Producers, consumers and decomposers, Energy flow in the ecosystem, Ecological succession, Food chains, food webs and ecological pyramids, Introduction, types, characteristic features, structure and function of the following ecosystems:

1. Forest ecosystem
2. Grassland ecosystem
3. Desert ecosystem
4. Aquatic ecosystems (ponds, streams, lakes, rivers, oceans, estuaries)

**Biodiversity and its conservation:** Introduction– Definition: genetic, species and ecosystem diversity. Biogeographical classification of India, Value of biodiversity: consumptive use, productive use, social, ethical, aesthetic and option values, Biodiversity at global, National and local levels, India as a mega-diversity nation, Hot-spots of biodiversity, Threats to biodiversity: habitat loss, poaching of wildlife, man-wildlife conflicts, Endangered and endemic species of India, Conservation of biodiversity: In-situ and Ex-situ conservation of biodiversity.

**Environmental Pollution:** Definition, Cause, effects and control measures of:

1. Air pollution
2. Water pollution
3. Soil pollution
4. Marine pollution
5. Noise pollution
6. Thermal pollution
7. Nuclear hazards

Solid waste Management: Causes, effects and control measures of urban and industrial wastes. Role of an individual in prevention of pollution. Pollution case studies. Disaster management: floods, earthquake, cyclone and landslides.

**Social Issues and the Environment:** From Unsustainable to Sustainable development, Urban problems related to energy, Water conservation, rain water harvesting, watershed management, Resettlement and rehabilitation of people; its problems and concerns-case studies, Environmental ethics: Issues and possible solutions. Climate change, global warming, acid rain, ozone layer depletion, nuclear accidents and holocaust- case studies. Wasteland reclamation, Consumerism and waste products, Environment Protection Act, Air (Prevention and Control of Pollution) Act, Water (Prevention and control of Pollution) Act, Wildlife Protection Act, Forest Conservation Act, Issues involved in enforcement of environmental legislation, Public awareness.

Human Population and the Environment: Population growth, variation among nations. Population explosion – Family Welfare Programme. Environment and human health. Human Rights. Value Education. HIV/AIDS.

Women and Child Welfare. Role of Information Technology in Environment and human health-case studies.

**Field work:**

1. Visit to a local area to document environmental assets: river/ forest/ grassland/ hill/ mountain.
2. Visit to a local polluted site-Urban/ Rural/ Industrial/ Agricultural.
3. Study of common plants, insects, birds.
4. Study of simple ecosystems-pond, river, hill slopes, etc.

**Text Books:**

1. *Textbook for Environmental Studies for Undergraduate Courses of all Branches of Higher Education*, Erach Bharucha for University Grants Commission, 2004.

**Reference Books:**

1. Trivedi R. K. and P.K. Goel, *Introduction to Air Pollution*, Techno-Science Publication.
2. Miller T.G. Jr. *Environmental Science*, Wadsworth Publishing Co.
3. Townsend C., Harper J, and Michael Begon, *Essentials of Ecology*, Blackwell Science.
4. Mhaskar A.K., *Matter Hazardous*, Techno-Science Publication.

Course Code	Course Name	L-T-P: C
MA201	Mathematics-III (Probability and Statistics)	3-1-0: 4

**Objectives:** Students will learn fundamental rules of Probability, discrete and continuous distributions, and statistical methods most commonly used in Computer Science and Software Engineering. They will be introduced to stochastic processes, Markov chains, statistical inference, and Monte Carlo methods and will apply the theory and methods to the evaluation of queuing systems and computation of their vital characteristics.

**Learning Outcomes:** On successful completion of this course, students should be able to:

1. Appreciate the significance of probability and statistics in computer science and related research areas.
2. Develop algorithmic approaches to real-life problems incorporating randomness in data.
3. Use appropriate statistical methods in the analysis of datasets.

**Prerequisite:** None

**Contents:**

**Module I:** Introduction: Classical, relative frequency and axiomatic definitions of probability, addition rule and conditional probability, multiplication rule, total probability, Bayes’ theorem and independence.

**Module II:** Random Variables and Distributions: Discrete, continuous and mixed random variables, probability mass, probability density and cumulative distribution functions, mathematical expectation, moments, probability and moment generating function, median and quantiles, Markov inequality, Chebyshev’s inequality, well-known discrete and continuous distributions.

**Module III:** Joint Distributions: Joint, marginal and conditional distributions, product moments, correlation and regression, independence of random variables, bivariate normal distribution.

**Module IV:** Transformations: functions of random vectors, distributions of order statistics, distributions of sums of random variables.

**Module V:** Sampling Distributions: Mean, median, variance, standard deviation, The Central Limit Theorem, distributions of the sample mean and the sample variance for a normal population, Chi-Square, t and F distributions.

**Module VI:** Stochastic Processes: Definition, characterization and types of random processes, WSS, SSS, ergodic, Gaussian random processes, AWGN model, Markov processes and Markov chains, counting processes, simulation of stochastic processes.

**Module VII:** Estimation and Hypothesis Testing: Unbiasedness, consistency, the method of moments and the method of maximum likelihood estimation, confidence intervals for parameters in one sample and two sample problems of normal populations, tests for one sample and two sample problems for normal populations, proportions, Chi square goodness of fit test.

**Text Books:**

1. Probability and Statistics for Computer Scientists, Michael Baron, CRC Press, 2ed, 2013.

**Reference Books:**

1. Probability & Statistics for Engineers & Scientists, Sharon L. Myers, Raymond Myers, Ronald Walpole, Keying Ye, 9th ed, Pearson, 2021.
2. Probability, Random Variables, and Stochastic Processes’, Athanasios Papoulis and S. Unnikrishna Pillai, Fourth Edition, McGraw-Hill, Europe, 2002.

3. Probability, Statistics, and Random Processes for Electrical Engineering', Alberto Leon-Garcia, Third Edition, Pearson, 2008.

Course Code	Course Name	L-T-P: C
HS201	Technical Writing	1-1-2: 3

**Objectives:** Provides training of students in developing skills required for developing technical documents through audience analysis, selecting suitable format, and the use of appropriate media. Improves professional communication with the use of rhetoric strategies and formal writing development. Builds expertise in the production of a variety of documents: memos, letters, proposals, and reports, while enhancing oral presentation skills and developing problem-solving skills with case study techniques to provide effective communication in the workplace.

**Learning Outcomes:** On successful completion of the course, students will be able to:

1. Edit prose for grammatical correctness, accuracy, diction, and audience-appropriate style.
2. Identify and select types of writing required in a variety of professional careers.
3. Assess the effectiveness and validity of information sources, including websites, business documents, and professional journals.
4. Determine writing objectives and compose and revise computer-based documents using appropriate formats and language.
5. Demonstrate effective communication principles practiced by professional writers.
6. Select and integrate graphics effectively with written communication.
7. Differentiate between facts and inferences in professional writing.
8. Use basic sources and methods of research and documentation in technology-related topics, including online research.
9. Summarize larger texts clearly and concisely for practical applications.

**Contents:**

**Module I:** Various Constructions: Structure of sentences, paragraphs, and documents, Stress for emphasis, Sequencing topics to create forward flow, Case Study 1: Comparison of poorly written vs well-structured software documentation, focusing on clarity, organization, and user understanding, Case Study 2: Analysis of how unclear problem descriptions and reports create confusion during debugging and delay issue resolution, Case Study 3: Examination of how ambiguous requirement specifications lead to

misinterpretation, inefficient workflows, and project delays in software development.

**Module II:** Style Manuals: Correct formats, Recording information, Citations (Chicago, APA & MLA Style)

**Module III:** Creative Writing: Use of literary tropes, Narrative craft and character development, Figures of speech, Writing for the reader.

**Module IV:** Technical Documents: Introduction to typesetting (LaTeX), Formats of technical documents, Experimental reports, technical reports, and proposals, Workshop on published documents, Discussion and workshop on term paper proposals.

**Module V:** Data Representation: Graphics and visual communication, Emphasis without distortion, Visual illusions, Minimalist approach to data representation, Univariate and multivariate displays

**Module VI:** Discussion and Workshop: Term papers, Elements of oral presentations, Oral presentations.

**Reference Books:**

1. Strunk, W., & White, E. B. (1972). The elements of style. Macmillan.
2. Perelman, L. (1998). The Mayfield handbook of technical and scientific writing. Mayfield Publishing Company.
3. Gopen, G. D., & Swan, J. A. (1990). The science of scientific writing. American Scientist, 78(6), 550–558.
4. Runeson, P., Höst, M., Rainer, A., & Regnell, B. (2012). Case study research in software engineering: Guidelines and examples. John Wiley & Sons. <https://doi.org/10.1002/97811181181034>

**Supplementary Readings:**

1. Markel, M. (2012). Technical communication. Bedford/St. Martin's.
2. Kent, P. (1998). Making money in technical writing. Macmillan USA.

Course Code	Course Name	L-T-P: C
CS201	Object Oriented Design & Programming	3-0-0: 3

**Objectives:** This course introduces basic concepts of object-oriented programming principles, design techniques, and analysis tools.

**Learning Outcomes:** On completion of this course the student should be able to design and analyze real-world problems based on object-oriented principles.

**Prerequisite:** CS101, CS161, CS102.

**Contents:**

**Module 1:** Introduction: Principles of OOD; programming Paradigms; benefits of OOD&P, applications of OOD; Classes and objects; access qualifiers; instance creation; constructors, parameterized constructors, overloaded constructors, constructors with default arguments, copy constructors, static class members, and static objects.

**Module II:** Functions and Operators: Function prototyping, function components, passing parameters, inline functions, default arguments, overloaded function; array of objects, pointers to objects, dynamic allocation operators, dynamic objects; Operator overloading, overloading a unary and binary operator, overloading the operator using friend function, stream operator overloading, data conversion.

**Module III:** Inheritance: Defining derived classes, single inheritance, protected data with private inheritance, multiple inheritance, multi-level inheritance, hierarchical inheritance, hybrid inheritance, multi-path inheritance, constructors in derived and base class, abstract classes, virtual function and dynamic polymorphism, virtual destructor.

**Module IV:** Exception Handling: Principle of exception handling, exception handling mechanism, multiple catch, nested try, re/throwing the exception.

**Text Books:**

1. Introduction to object-oriented programming, B. Timothy, Pearson, 2001.
2. Object Oriented Design and Patterns, C. Horstmann, John Wiley & Sons, 2005.

Course Code	Course Name	L-T-P: C
CS261	Object Oriented Design & Programming Lab	0-0-2: 1

**Objectives:** To provide hands-on experience in object-oriented programming concepts such as classes, inheritance, polymorphism, operator overloading, and exception handling through practical implementation.

**Learning Outcomes:** Students will be able to implement OOP concepts, design class-based programs, apply inheritance and polymorphism, and handle memory and exceptions effectively.

**Tentative list of Lab exercises**

**Lab Program 1** introduces the fundamentals of Object-Oriented Design by implementing classes and objects, using access specifiers, and writing constructors including default, parameterized, and copy constructors.

**Lab Program 2:** focuses on creating static class members and static objects, and understanding their their scope and behavior in object-oriented programming.

**Lab Program 3:** emphasizes function prototyping, passing parameters, using inline functions, default arguments, and function overloading to enhance code reusability.

**Lab Program 4:** involves creating arrays of objects and dynamic memory allocation using pointers and the new/delete operators to manage object lifetimes.

**Lab Program 5:** demonstrates operator overloading by overloading unary and binary operators, including stream operators for input/output operations.

**Lab Program 6:** focuses on implementing friend functions for operator overloading and practicing type conversion between class and primitive data types.

**Lab Program 7:** covers inheritance by creating programs involving single, multilevel, and hierarchical inheritance and observing constructor invocation order.

**Lab Program 8:** explores advanced inheritance types such as multiple, hybrid, and multipath inheritance, and introduces abstract classes and virtual functions to demonstrate dynamic polymorphism.

**Lab Program 9:** includes writing programs that use virtual destructors and handle object destruction correctly in polymorphic scenarios.

**Lab Program 10:** introduces exception handling by writing programs using try-catch blocks, handling multiple and nested exceptions, and demonstrating rethrowing mechanisms.

Course Code	Course Name	L-T-P: C
CS203	Design and Analysis of Algorithms	3-0-0: 3

**Objectives:** This course introduces fundamental techniques for designing and analyzing algorithms, focusing on efficiency, key design paradigms, and problem-solving skills for real-world computational challenges.

**Learning Outcomes:** After completing this course, students will be able to:

1. Analyze the time and space complexity of algorithms using asymptotic techniques.
2. Apply divide and conquer, greedy, and dynamic programming approaches to solve problems.
3. Implement and evaluate sorting and searching algorithms.

4. Design and analyze graph algorithms such as shortest paths, spanning trees, and flows.
5. Understand and apply advanced concepts like amortized analysis and randomized algorithms.
6. Solve real-world problems using efficient algorithmic strategies.

**Contents:**

**Module I:** Foundations: Asymptotic notation, Divide and Conquer Techniques, Solving Recurrences, Probabilistic Analysis and Randomized Algorithms

**Module II:** Sorting and Order Statistics: Heapsort, Quicksort, Sorting in Linear Time, Median Order Statistics.

**Module III:** Advanced Design and Analysis Techniques: Dynamic Programming, Greedy Algorithms and Amortized Analysis

**Module IV:** Graph Algorithms: Depth First search, Breadth First Search, Bi-connectivity and Strong Connectivity, Topological Sort, Minimum Spanning Trees, Shortest Paths and Maximum Flow

**Module V:** String Matching: Rabin-Karp and Knuth-Morris-Pratt’s string matching algorithms. P, NP, NP-hard, and NP Complete problems.

**Text Books:**

1. Introduction to Algorithms by Cormen, Lieserson, Rivest and Stein, 2nd or 3rd edition, Eastern Economy Edition, MIT Press.

Course Code	Course Name	L-T-P: C
CS263	Design and Analysis of Algorithms Lab	0-0-1: 1

**Objectives:** To provide hands-on implementation of algorithm design techniques and analyze their efficiency for solving computational problems.

**Learning Outcomes:** Students will be able to implement and analyze algorithms, apply design paradigms, and solve problems using efficient techniques.

**Tentative list of Lab Programs:**

1. Implementation of asymptotic analysis and comparison of algorithm complexities.
2. Divide and Conquer: Merge Sort / Quick Sort.
3. Heap Sort and priority queue operations.
4. Linear time sorting (Counting/Radix Sort).
5. Dynamic Programming: Knapsack / Matrix Chain Multiplication.
6. Greedy Algorithms: Activity Selection / Huffman Coding.

7. Graph Traversal: BFS and DFS.
8. Minimum Spanning Tree: Prim’s / Kruskal’s Algorithm.
9. Shortest Path: Dijkstra / Bellman-Ford Algorithm.
10. String Matching: Rabin-Karp / KMP Algorithm.

Course Code	Course Name	L-T-P: C
EC201	Digital Logic Design	3-0-0: 3

**Objectives:** The main objective of this course is to introduce the concept of digital logic circuits and binary systems. To design and implement the combinational and sequential logic circuits which are the prime ingredient to make ALU. To provide the basic understanding of design and implementation of the digital circuits using VHDL.

**Learning Outcomes:** After completing this course, students will be able to:

1. Understand the fundamentals of digital logic design.
2. Understand the combinational logic circuits.
3. Understand the sequential logic circuits.
4. Understand the design and implementation of digital circuits using VHDL.

**Prerequisite:** Boolean Algebra and Basics of Semiconductor

**Contents:**

**Module I:** Numbers systems and Boolean algebra: Introduction to number systems and Boolean algebra; Boolean identities, basic logic functions, standard forms of logic expressions, simplification of logic expressions.

**Module II:** Combinational logic: Arithmetic circuits, decoders, encoders, multiplexers, de-multiplexers, and their use in logic synthesis; Hazards in combinational circuits.

**Module III:** Sequential logic circuits: Latches and Flip Flops (SR, D, JK, T); Timing in sequential circuits; Shift register; Counters – synchronous, asynchronous; Sequential circuit design examples in VHDL and simulation.

**Module IV:** Introduction to VHDL: Behavioral – data flow, and algorithmic and structural description, lexical elements, data objects types, attributes, operators; VHDL coding examples, combinational circuit design examples in VHDL and simulation.

**Module V:** Finite state machines: Basic concepts and design; Moore and Mealy machines examples; State minimization/reduction, state assignment; Finite state machine design case studies and FSM circuit design

examples in VHDL and simulation. ROM and RAM, PLA, PAL and FPGA; RTL based design projects and their implementation in FPGA using VHDL.

**Module VI:** Logic families: Brief overview of Transistor as a switch; Logic gate characteristics – propagation delay, speed, noise margin, fan-out and power dissipation; Standard TTL and static CMOS gates.

**Text Books:**

1. Digital Design, Morris Mano, Prentice Hall, 2002.
2. Digital Fundamentals, 10th Ed, Floyd T L, Prentice Hall, 2009.

**Reference Books:**

1. Digital Circuits and Design, S Salivahanan, Vikas Publication House Pvt. Ltd.
2. A VHDL Primer, 3rd Edition, J. Bhaskar, Pearson.
3. Digital Design-Principles and Practices, 4th Ed, J F Wakerly, Prentice Hall, 2006.
4. Pedroni V.A., “Digital Circuit Design with VHDL”, Prentice Hall India, 2nd 2001 Edition

Course Code	Course Name	L-T-P: C
EC261	Digital Logic Design Lab	0-0-2: 1

**Objectives:**

1. To familiarize the students with the basic concept of digital logic gates.
2. To explain the various digital logic hardware circuits.
3. To provide the hands-on experience of both digital software and hardware circuits.
4. To familiarize the students with the hardware description language (VHDL).

**Learning Outcomes:** After completing this course, students will be able to:

1. Understand the fundamentals of digital logic design.
2. Understand the combinational logic circuits.
3. Understand the sequential logic circuits.
4. Understand the design and implementation of digital circuits using VHDL.

**List of Experiments:**

**PART I:** Digital Design Hardware Implementation

1. Design and hardware implementation of: (a) 2-bit Adder/Subtractor with XOR as well as NAND gates, (b) 4:1 Multiplexer using universal gates and realization of Full Adder using Multiplexers, BCD Adder using two binary adders (IC based) and

other gates, 3:8 Decoder and realization of Full Adder.

2. Realization of R-S, D and J-K latches and D Flip-Flop.
3. Realization of Mod-8 Up-Down Ripple Counter.
4. Realization of synchronous Mod-3 and Mod-2 counters.
5. Realization of higher Mod counters by cascading lower Mod counters.
6. Design and hardware implementation Finite State Machine (simple).

**PART II:** Digital System Design using HDL and EDA

1. Modeling different types of gates:(a) 2-input NAND (b) 2-input OR gate (c) 2-input NOR gate (d) NOT gate (e) 2-input XOR gate (f) 2-input XNOR gate.
2. Modeling a Half-adder, Full-adder, D flip-flop and D Latch.
3. Modeling a 2-to-1 Multiplex and 2-to-4 Decoder.
4. Modeling a 4-to-1 Multiplexer and Modeling a 4-bit parallel adder.
5. Modeling a Debouncer (and stop-watch).
6. Modeling a Digital to Analog converter.

**Text Books:**

1. Digital Design, Morris Mano, Prentice Hall, 2002.
2. Digital Fundamentals, 10th Ed, Floyd T L, Prentice Hall, 2009.

**Reference Books:**

1. Digital Circuits and Design, S Salivahanan, Vikas Publication House Pvt. Ltd.
2. A VHDL Primer, 3rd Edition, J. Bhaskar, Pearson.
3. Digital Design-Principles and Practices, 4th Ed, J F Wakerly, Prentice Hall, 2006.
4. Pedroni V.A., “Digital Circuit Design with VHDL”, Prentice Hall India, 2nd 2001 Edition.

Course Code	Course Name	L-T-P: C
PH201	Optics	2-1-2: 4

**Objectives:** This course provides a foundational understanding of light propagation and optical principles in real-world applications. It develops analytical and experimental skills for solving optical engineering problems, prepares students for careers in photonics and imaging technologies, and builds a foundation for research in advanced optical and adaptive systems.

**Learning Outcomes:** This course will enable students to:

1. Understand the fundamental principles of ray, wave, and electromagnetic optics.
2. Apply concepts of diffraction, interference, and Gaussian beam propagation in the optical system.
3. Develop problem-solving skills for real-world optical systems.
4. Utilize their knowledge in industries such as photonics, imaging, free-space optical communication, medical optics, and optical system design.

**Prerequisites:** PH101, PH102

**Contents:**

**Module-I:** Ray Optics and Optical Systems: Laws of reflection and refraction; reflection and refraction by plane and spherical surfaces; optical path, refracting and reflecting surfaces under paraxial approximation; matrix formulation (ABCD matrices) for lenses, mirrors, and optical systems; image formation and magnification.

**Module-II:** Aberrations and Optical Design: Seidel's primary monochromatic aberrations; chromatic aberrations; methods of aberration correction; concept of aperture stop, field stop, and entrance/exit pupils; introduction to optical design principles in lens systems.

**Module-III:** Wave Optics and Interference: Nature of light as an electromagnetic wave; principle of superposition and coherence; two-beam and multiple-beam interference; Michelson and Fabry-Perot interferometers; applications of interference; fundamentals of holography and reconstruction of wavefronts.

**Module-IV:** Diffraction and Gaussian Beam Propagation: Fresnel and Fraunhofer diffraction; diffraction from single slit, rectangular and circular apertures; resolving power of optical instruments; Gaussian beams, their parameters and propagation characteristics through optical elements; optical Fourier transforms.

**Module-V:** Polarization and Adaptive Optics: Polarization of light—linear, circular, and elliptical; polarizing components including polarizers, retarders, and wave plates; fundamentals of adaptive optics; wavefront sensing methods (interferometric, Shack-Hartmann, and grating-array based sensors); wavefront correction and system reconstruction.)

**Text/Reference Books:**

1. Fundamentals of Optics: Jenkins and White, 4th Edition, McGraw-Hill College, 1976.
2. Principle of Optics, by Born and Wolf, 7th Edition, Cambridge University Press, 2001.
3. Optics, by Hecht, 4th Edition, Addison-Wesley, 2002.

4. Introduction to Optics, Pedrotti, Pedrotti, and Pedrotti, 3rd Edition, Cambridge University Press, 2001.
5. W. Goodman, Introduction to Fourier Optics, 3rd Edition, Robert and Company, 2005.

Course Code	Course Name	L-T-P: C
AI201	Operating Systems	3-0-0: 3

**Objective:** This course provides undergraduate students with knowledge about contemporary operating system design, its relationship between memory, processor, file interface, and processes, with exposure to Linux-based systems and security aspects of operating systems.

**Learning Outcomes:** On successful completion of this course, students should be able to:

1. Describe process management and concepts of threading, multitasking, IPC, and Linux process handling.
2. Differentiate scheduling algorithms and identify causes of deadlocks and their remedial measures, including security implications.
3. Describe memory management techniques, file system interfacing, and disk scheduling, with focus on protection and access control.
4. Understand OS components and analyze basic OS security mechanisms such as authentication, access control, and isolation

**Contents:**

**Module I - Introduction-** Overview and history of Operating Systems, functions of an OS, device drivers, I/O interrupts, system call interface, and introduction to Linux OS, shell basics, and security features of OS.

**Module II - Process Management-** Process abstraction, process states and PCB, threads, process scheduling, real-time scheduling, classical synchronization problems, semaphores, monitors, deadlocks and their handling, along with process isolation and security considerations in multitasking systems.

**Module III - Memory Management-** Segmentation, demand paging, hardware support, page fault handling, page replacement algorithms, shared memory, and memory protection mechanisms and secure memory management concepts.

**Module IV - Storage Management-** Disk structure, disk scheduling, disk management, RAID structure, file organization and mounting, file descriptors, directory structure and implementation, I/O systems, along with file system security, permissions (Linux), and access control mechanisms.

**Text Books:**

1. Operating Systems: Internals and Design Principles.
2. Modern Operating Systems

**Reference Books:**

1. Operating System Concepts.
2. Operating System Concepts Based Approach.

Course Code	Course Name	L-T-P: C
AI261	Operating Systems Lab	0-0-2: 1

**Tentative List of Lab Exercises**

1. Process creation and management using fork(), exec(), wait() in Linux.
2. Implementation of CPU scheduling algorithms (FCFS, SJF, Round Robin).
3. Synchronization problems using semaphores/mutex (Producer-Consumer, Reader-Writer).
4. Thread creation and synchronization using pthreads.
5. Deadlock detection and recovery simulation.
6. Implementation of page replacement algorithms (FIFO, LRU, Optimal).
7. Simulation of demand paging and page fault handling.
8. Linux file permissions and access control (chmod, chown, ACLs).
9. Basic security exercises: process monitoring, user management, and system protection in Linux

Course Code	Course Name	L-T-P: C
AI203	Linear Systems and and Signal Processing	2-0-2: 3

**Description:** This course provides a mathematical foundation for the analysis of signals and systems in both continuous and discrete time. It bridges the gap between physical phenomena (signals) and the mathematical abstractions (systems) used to process them. The course emphasizes the transformation of signals between time and frequency domains, essential for applications in communications, control, and image processing.

**Objectives:** The primary objective of this course is to equip students with the analytical tools necessary to characterize linear time-invariant (LTI) systems and evaluate their response to various signal inputs. Students will learn to apply Fourier, Laplace, and

Z-transforms to solve differential and difference equations while developing a deep understanding of the sampling theorem, which enables the interface between analog and digital worlds.

**Learning Outcomes:** Upon completion, students will be able to:

1. Classify signals and systems based on their mathematical properties and compute the output of LTI systems using convolution.
2. Demonstrate proficiency in analyzing frequency-domain characteristics through spectral analysis and effectively use transform techniques to determine system stability and causality.
3. Gain hands-on experience in implementing these concepts using software tools like MATLAB or Python in a laboratory setting.

**Prerequisite:** None

**Contents:**

**Module I:** Introduction to Signals and Systems Classification of signals: continuous-time and discrete-time, periodic and aperiodic, even and odd, energy and power signals. Basic operations on signals: shifting, scaling, and reversal. Classification of systems: linearity, time-invariance, causality, stability, and invertibility. Mathematical representation of LTI systems using impulse response and the convolution integral/sum.

**Module II:** Fourier Analysis of Continuous-Time Signals Representation of periodic signals using the Trigonometric and Exponential Fourier Series. Convergence conditions and properties of Fourier Series. Extension to aperiodic signals via the Continuous-Time Fourier Transform (CTFT). Frequency response of LTI systems and the concept of filtering (Low-pass, High-pass, and Band-pass).

**Module III:** Sampling and Discrete-Time Analysis The Sampling Theorem and the Nyquist rate. Effects of undersampling and Aliasing. Reconstruction of signals using interpolation. Introduction to the Discrete-Time Fourier Transform (DTFT). Analysis of discrete-time LTI systems and their representation through difference equations.

**Module IV:** Transform Domain Analysis (Laplace and Z-Transform) Definition of the Laplace Transform and its Region of Convergence (ROC). Solving differential equations and analyzing system stability using pole-zero plots. Introduction to the Z-Transform for discrete-time signals. Relationship between the Laplace, Fourier, and Z-transforms in the context of system transfer functions.

**Laboratory:**

1. Introduction to MATLAB/Python for signal generation.
2. Implementation of Linear and Circular Convolution.
3. Computation of Fourier Series coefficients for square and triangular waves.
4. Analyzing frequency response and Designing basic Digital Filters.
5. Simulation of Sampling and Reconstruction to observe Aliasing.

**Text Books:**

1. Oppenheim, A. V., Willsky, A. S., & Nawab, S. H. (1997). Signals and Systems (2nd ed.). Pearson Education. ISBN: 978-0138147570.
2. Lathi, B. P., & Green, R. A. (2018). Linear Systems and Signals (3rd ed.). Oxford University Press. ISBN: 978-0190200176.

**Reference Books:**

1. Haykin, S., & Van Veen, B. (2007). Signals and Systems (2nd ed.). Wiley. ISBN: 978-0471138204.
2. Hsu, H. P. (2013). Schaum's Outline of Signals and Systems. McGraw-Hill. ISBN: 978-0071829465.

Course Code	Course Name	L-T-P: C
CC201	Networks Fundamentals	3-0-2: 4

**Objectives:** Provide essential knowledge about computer networks, as well as related concepts and theories.

**Learning Outcomes:** On successful completion of the course students will be able to:

1. Gain understanding of fundamental network theory and layered communication architectures.
2. Provide solutions to various challenges in network theory.
3. Appreciate the layered model of computer networking.
4. Identify basic protocols and design issues related to the layered model.
5. Design and implement protocols for various networking layers.

**Prerequisite:** None

**Contents:**

**Module I:** Application Layer- This module covers the principles of network applications, including client-server and peer-to-peer models, process communication, and socket programming. It introduces key application protocols such as HTTP, FTP, SMTP,

POP3/IMAP, and DNS, along with an overview of multimedia and peer-to-peer applications.

**Module II:** Transport Layer- This module focuses on transport services, multiplexing and demultiplexing, and reliable data transfer. It explains UDP and TCP, including flow control, congestion control, and TCP congestion control mechanisms.

**Module III:** Network Layer- This module introduces network layer functions such as forwarding and routing. It covers routing algorithms, IP addressing (IPv4/IPv6), subnetting, ICMP, DHCP, and an overview of Internet routing and software-defined networking.

**Module IV:** Data Link Layer- This module discusses data link layer services, error detection and correction, and multiple access protocols such as ALOHA and CSMA/CD. It also includes Ethernet, ARP, LAN switching, VLANs, and wireless LANs.

**Module V:** Physical Layer and Network Security- This module presents basic concepts of the physical layer, transmission media, and signal transmission. It also introduces network security concepts including cryptography, authentication, SSL/TLS, firewalls, intrusion detection systems, and common attacks with countermeasures.

**Text Books:**

1. Computer Networking: A top-down approach featuring the Internet-J. F. Kurose, K. W. Ross.
2. Computer Networks, A. S. Tanenbaum.
3. Computer Networks: A Systems Approach, L. L. Peterson, B. S. Davie

**Reference Books:**

1. AS Tanenbaum, DJ Wetherall, Computer Networks, 5th Ed., Prentice-Hall, 2010.
2. LL Peterson, BS Davie, Computer Networks: A Systems Approach, 5th Ed., Morgan-Kaufman, 2011.
3. W Stallings, Cryptography and Network Security, Principles and Practice, 5th Ed., Prentice-Hall, 2010.

Course Code	Course Name	L-T-P: C
EC203	Linear Systems and Signal Processing	2-0-2: 3

**Objectives:** The objective of this course is to provide a comprehensive understanding of continuous-time and discrete-time signals and systems, including their classification and fundamental properties. It develops the ability to analyse Linear Time-Invariant (LTI) systems using time-domain methods such as

convolution and differential or difference equations. The course also introduces frequency-domain analysis through Fourier Series and Fourier Transform, enabling deeper insight into system behaviour. Further, it equips students with essential mathematical tools such as the Laplace Transform and Z-Transform for efficient system analysis and problem-solving. Additionally, the course covers sampling theory, including aliasing and signal reconstruction, forming a foundation for digital signal processing. Overall, it prepares students to apply these concepts in practical engineering domains such as communication, control, and signal processing systems.

### Learning Outcomes:

1. Understand and classify continuous-time and discrete-time signals and systems.
2. Analyse properties of systems such as linearity, time-invariance, causality, and stability.
3. Apply convolution to determine the output of Linear Time-Invariant (LTI) systems.
4. Represent and analyse signals using the Fourier Series and Fourier Transform.
5. Use the Laplace Transform to analyse continuous-time systems and determine system behaviour.
6. Apply Z-Transform for analysis of discrete-time systems.
7. Determine system characteristics using poles, zeros, and transfer functions.
8. Understand and apply sampling theorem, aliasing, and signal reconstruction concepts.
9. Solve engineering problems related to communication, control, and signal processing.
10. Develop analytical and problem-solving skills for real-world signal and system applications.

**Prerequisite:** Linear Algebra

### Contents:

**Module I:** Signals and Basic Operations: Definition and classification of signals, Standard signals (Unit step, impulse, ramp, exponential, sinusoidal), Signal operations (Time shifting, scaling, reversal), Basic system properties (Linearity, time-invariance, causality, stability)

**Module II:** System Representation and Convolution: Continuous-time and discrete-time systems, Impulse response and system characterization, Convolution (Continuous vs Discrete time system), Properties of convolution, Interconnection of systems and equivalent system realization

**Module III:** Fourier Series and Fourier Transform: Fourier Series and its properties (Trigonometric and exponential forms), Fourier Transform (CTFT and DTFT), Properties of Fourier Transform (linearity,

shifting, modulation, convolution), Frequency response of systems, Parseval's theorem

**Module IV:** Laplace Transform: Definition and region of convergence (ROC), Properties of Laplace Transform, Inverse Laplace Transform, System analysis using Laplace Transform, Transfer function and poles & zeros, Stability analysis

**Module V:** Z-Transform and Sampling: Z-Transform, Definition and ROC, Properties of Z-Transform, Inverse Z-transform, System analysis using Z-transform, Sampling theorem, Nyquist rate, Aliasing, Reconstruction of signals and FFT

### Text Books:

1. Oppenheim, Willsky & Nawab — "Signals and Systems" (2nd Ed.).
2. B.P. Lathi — "Principles of Signal Processing and Linear Systems" (2nd Ed.)

### Reference Books:

1. Simon Haykin & Barry Van Veen — "Signals and Systems" (2nd Ed.)
2. Michel J. Roberts "Fundamentals of Signals and Systems"

### List of Experiments:

1. Standard Signals: Generate unit step, impulse, ramp, exponential, and sinusoidal signals in MATLAB. Plot and compare continuous vs. discrete versions using stem and plot.
2. Signal Operations: Apply time shifting, scaling, and reversal to standard signals. Test basic system properties: linearity, time-invariance, causality via input-output pairs.
3. Convolution Basics: Compute discrete convolution, visualize sliding window. Verify commutativity and associativity properties analytically and graphically.
4. System Interconnections: Simulate series, parallel, and feedback interconnections. Derive and plot equivalent impulse responses for continuous/discrete LTI systems.
5. Plot the impulse response, and step response of the RLC circuit. Obtain the magnitude and phase response of the RLC circuit.
6. Fourier Series: Synthesize periodic signals using trigonometric and exponential Fourier series.
7. Fourier Transform: Compute CTFT/DTFT pairs; explore linearity, shifting, modulation properties. Plot magnitude/phase of frequency responses and verify Parseval's theorem.
8. Laplace & Stability: For Laplace transforms, ROC, inverses, and transfer functions. Plot poles/zeros; analyze stability via pole locations.

9. Z-Transform & Sampling: Implement Z-transforms, inverses; design simple filters. Demonstrate sampling theorem, aliasing, Nyquist rate, reconstruction, and FFT spectra.

Course Code	Course Name	L-T-P: C
EC205	Antenna and Wave Propagation	3-0-0: 3

**Objectives:** The objective of this course is to study various wave propagation mechanisms, including ground wave, skywave, and space wave propagation, and understand their significance in communication systems. It aims to analyze array configurations and beamforming techniques while exploring their applications in modern communication networks. Students will apply theoretical concepts to real-world scenarios in wireless communication, satellite technology, and radar systems. The course also focuses on advancements in antenna technologies, such as smart antennas and MIMO systems, and their impact on communication efficiency. Ultimately, it equips students with the knowledge and skills necessary to analyze, design, and implement antenna systems for various communication applications.

#### Learning Outcomes:

1. Understand Electromagnetic Wave Propagation – Explain the fundamental principles of electromagnetic waves and their propagation characteristics.
2. Design and Analyze Antenna Parameters – Evaluate key antenna parameters such as gain, directivity, efficiency, polarization, and radiation patterns. Develop and assess different types of antennas, including wire antennas, aperture antennas, and microstrip antennas.
3. Examine Wave Propagation Mechanisms – Understand and analyze different propagation modes, including ground wave, skywave, space wave, and tropospheric propagation.
4. Implement Antenna Systems in Communication – Apply knowledge of antennas and wave propagation in practical communication systems such as satellite communication, radar, and wireless networks.
5. Solve Real-World Engineering Problems – Utilize theoretical concepts to solve practical engineering challenges related to antennas and wave propagation.

**Prerequisite:** Electromagnetic Theory, Signals and Systems, Basic Communication System.

#### Contents:

**Module-I:** Review of Maxwell equations, retarded potentials, Hertzian dipole, antenna parameters: radiation pattern, near- and far-field regions, reciprocity, directivity and gain, effective aperture, polarization, input impedance, efficiency, Friis transmission equation, radiation integrals and auxiliary potential functions.

**Module-II:** Infinitesimal dipole, finite-length dipole, monopole, linear elements near or on infinite conductors, dipoles for mobile communication, small circular loop, method of moments and its application to wire antenna or scatterer, S-parameters, Smith chart.

**Module-III:** Arrays of point sources, broadside, endfire, and parasitic arrays, extension to planar arrays, synthesis of antenna arrays using Schelkunoff polynomial method, and Fourier transform method, Multi-input multi-output antennas, vivaldi antenna.

**Module-IV:** Huygens' principle, radiation from rectangular and circular apertures, design considerations, Babinet's principle, Fourier transform method in aperture antenna theory, slot and horn antennas, parabolic reflector antennas, broadband antennas, introduction to microstrip antenna and its feeding techniques.

**Module-V:** Different modes of wave propagation, ground wave propagation, space wave propagation, scattering phenomena, skin depth, wave propagation inside the materials, introduction to electromagnetic absorbers, introduction to reconfigurable intelligent surfaces, antenna measurements: test ranges, radiation pattern, directivity, gain and impedance measurement, near field and far field techniques.

#### Text Books:

1. Balanis, C.A., "Antenna Theory and Design", 3rd Ed., John Wiley & Sons.
2. Jordan, E.C. and Balmain, K.G., "Electromagnetic Waves and Radiating Systems", 2nd Ed., Prentice-Hall of India.
3. Stutzman, W.L. and Thiele, H.A., "Antenna Theory and Design", 2nd Ed., John Wiley & Sons.

#### Reference Books:

1. Elliot, R.S., "Antenna Theory and Design", Revised edition, Wiley IEEE Press.
2. Garg, R., Bhartia, P., Bahl, I. and Ittipiboon, A., "Microstrip Antenna Design Handbook", Artech House.

Course Code	Course Name	L-T-P: C
EC263	Antenna and Wave Propagation Lab	0-0-2: 1

**Objectives:** The "Antenna and Wave Propagation Lab" course provides hands-on experience with antennas, enabling students to design, construct, and test various antenna types while learning to measure critical antenna parameters such as radiation patterns, gain, impedance, and bandwidth. Through wave propagation experiments, students analyze how electromagnetic waves behave in different environments, including the effects of reflection, diffraction, and scattering. The course also emphasizes familiarity with lab equipment, teaching students to operate tools like network analyzers, signal generators, and spectrum analysers, while incorporating simulation and modelling using software like HFSS or CST to predict and validate antenna performance. Additionally, students gain an understanding of antenna arrays and beamforming, exploring how phased arrays and beam steering are used in modern communication systems. Finally, the course focuses on real-world applications, allowing students to apply their knowledge to practical scenarios such as designing antennas for wireless communication, and satellite systems, and analysing propagation challenges in urban and rural environments.

**Laboratory Outcomes:**

1. Dipole Antenna Design and Radiation Pattern Measurement.
2. Measurement of Antenna Gain and Directivity.
3. VSWR and Impedance Matching of Antennas.
4. Study of Microstrip Patch Antennas.
5. Measurement of Ground Wave and Sky Wave Propagation.
6. Simulation of Antenna Arrays and Beamforming.
7. Analysis of Multipath Effects in Urban Environments.
8. Measurement of Polarization and Its Impact on Antenna Performance.

**List of Laboratory Experiments:**

**Hardware based experiments:**

1. Gain and directivity measurement of various antennas.
2. S-parameter analysis of various antennas (Dipole, Folded Dipole, Horn, and Yagi-Uda).
3. VSWR and impedance matching of antennas.
4. Fabricate the designed microstrip antenna and compare the performance with simulated results.
5. Fabricate and test the designed antenna array and study the beamforming.

**Simulation based experiments:**

1. Designing a rectangular or circular patch antenna for a specific frequency and analyzing the effect of substrate permittivity.
2. Optimization of spacing and length of each element in a parasitic array.

3. Design and simulation of 2X2 or 4X4 microstrip patch array and observe the impact of mutual coupling between elements.
4. Analyse the performance differences between probe feed, microstrip line feed, and aperture coupled feed for patch antennas.
5. Design a broadband antenna for wireless application.

**Text Books:**

1. Balanis, C.A., "Antenna Theory and Design", 3rd Ed., John Wiley & Sons.
2. Jordan, E.C. and Balmain, K.G., "Electromagnetic Waves and Radiating Systems", 2nd Ed., Prentice-Hall of India.
3. Stutzman, W.L. and Thiele, H.A., "Antenna Theory and Design", 2nd Ed., John Wiley & Sons.

**Reference Books:**

1. Elliot, R.S., "Antenna Theory and Design", Revised edition, Wiley IEEE Press.
2. Garg, R., Bhartia, P., Bahl, I. and Ittipiboon, A., "Microstrip Antenna Design Handbook", Artech House.

Course Code	Course Name	L-T-P: C
EC207	Network Analysis and Synthesis	2-0-0: 2

**Objectives:** To develop an in-depth understanding of network analysis techniques and synthesis methodologies. This course covers fundamental concepts such as circuit theorems, network topology, time and frequency domain analysis, and network synthesis. It emphasizes practical problem-solving and computational approaches for electrical networks.

**Learning Outcomes:** On successful completion of this course, students should be able to:

1. Understand fundamental principles and techniques of network analysis.
2. Analyze electrical circuits using mesh, node, Laplace transforms, and state-variable methods.
3. Perform time-domain and frequency-domain analysis of RL, RC, and RLC circuits.
4. Apply network synthesis techniques and design active/passive filters using computational tools.

**Prerequisite:** Basic electrical circuit concepts, Ohm's law, and an introduction to differential equations and Laplace transforms.

**Contents:**

**Module-I:** Network Theorems and Topology: Node and mesh analysis, wye-delta transformation, superposition

theorem, Thevenin's theorem, Norton's theorem, reciprocity, and maximum power transfer theorem. Graph theory: incidence and adjacency matrices, trees, cut-sets, tie-sets.

**Module-II:** Time and Frequency Domain Analysis: Sinusoidal steady-state analysis: phasors, complex power. Time and frequency domain analysis of RL, RC, and RLC circuits. Solution of network equations using Laplace transforms – introduction to state-variable analysis.

**Module-III:** Two-Port Networks and Filters: Linear 2-port network parameters: Z, Y, h, and ABCD. Interrelationship between two-port parameters, reciprocity, and transmission properties. Introduction to passive and active filter design.

**Module-IV:** Network Synthesis: Poles and zeros of network functions, Synthesis of LC, RC, and RL networks. Positive real functions and their properties. Approximation methods for network synthesis.

#### Text Books:

1. Hayt, Kemmerly and Durbin, "Engineering Circuit Analysis", 8th Ed., Tata McGraw-Hill (2012).
2. M.E. Van Valkenburg, "Network Analysis", 3rd Ed., Pearson.

#### Reference Books:

1. S. Ghosh, Network Theory: Analysis and Synthesis, PHI Learning.
2. Kuo, F.F., "Network Analysis and Synthesis", 2nd Ed., Wiley India (2008).

Course Code	Course Name	L-T-P: C
MC201	Calculus of Several Variables	3-1-0: 4

**Objectives:** This course aims to provide a solid foundation in the topology of Euclidean spaces, multivariable calculus, and vector analysis. It focuses on concepts of compactness and continuity, differentiability and optimization of multivariable functions, and integral theorems such as Green's, Stokes', and Gauss' theorems, emphasizing both theory and applications.

**Learning Outcomes:** After completing this course, students will be able to:

1. Understand and apply topological concepts such as compactness and separation in  $\mathbb{R}^n$  under various norms.
2. Analyze and compute derivatives of multivariable functions, and apply the contraction principle, inverse, and implicit function theorems.

3. Use Taylor's theorem and the method of Lagrange multipliers to solve optimization problems.
4. Evaluate multiple integrals and apply Fubini's, Green's, Stokes', and Gauss' theorems to problems involving vector fields.

**Prerequisite:** Linear Algebra, Calculus.

#### Contents:

**Module 1:** Functions of several variables, continuity, differentiability, derivative of a function from  $\mathbb{R}^m$  to  $\mathbb{R}^n$ , contraction principle, inverse function theorem and implicit function theorems. Taylor's theorem, Lagrange multiplier's method.

**Module 2:** Integration on  $\mathbb{R}^n$ , Fubini's theorem, Change of variables, Stokes' theorem, vector fields, gradient, curl, divergence, Green's theorem, Gauss's (divergence) theorem.

**Module 3:** Metric topology of  $\mathbb{R}^n$ , topology induced by  $L_p$ -norms and their equivalences, compactness, separation properties.

#### Text Books:

1. Walter Rudin, Principles of Mathematical Analysis, McGraw-Hill, 1986.
2. Tom M. Apostol, Mathematical Analysis, 2nd Edition, Narosa Publishing House, 1974.

#### Reference Books:

1. Tom M. Apostol, Calculus, Vol. II: Multi-Variable Calculus and Linear Algebra with Applications to Differential Equations and Probability, 2nd Edition, Wiley, 1969.
2. H.L. Royden and P.M. Fitzpatrick, Real Analysis, 4th Edition, Pearson, 2010.
3. Michael Spivak, Calculus on Manifolds: A Modern Approach to Classical Theorems of Advanced Calculus, Addison-Wesley, 1965.
4. K.A. Ross, Elementary Analysis: The Theory of Calculus, Springer, 1980.

## Semester-IV

### Courses and Contents

Course Code	Course Name	L-T-P: C
MA202	Numerical Techniques	0-1-2: 2

**Objectives:** This course provides insights into implementation of numerical computing methods that are practical, efficient, and elegant.

**Learning Outcomes:** On successful completion of this course, students should be able to:

1. Implement numerical algorithms for specific problems.
2. Apply an algorithm with analysis of cost benefits.

**Prerequisites:** Linear Algebra, Calculus.

**Contents:** List of Experiments:

1. Solving systems of linear algebraic equations: Direct and Iterative Methods, Singular Value Decomposition, Ill-Conditioned System.
2. Eigen Value Computation: Jacobi Method for Symmetric Matrices.
3. Solving scalar nonlinear equations: Bisection Method, Newton-Raphson Method, Secant Method, Regula-Falsi method.
4. Interpolation and Curve Fitting: Polynomial interpolation, Extrapolation, Cubic Spline, Least Square fit.
5. Numerical Differentiation: Finite Difference Approximation, Derivatives by Interpolation (Using Forward/ Backward/central difference formula)
6. Numerical Integration: Trapezoidal and Simpson's rules for integration.
7. Solution of initial value problems: Euler's method, Euler's modified method, Runge-Kutta methods, Milne PC method.
8. Finite Difference Schemes for boundary value problems: Second-order ordinary differential equation with boundary conditions.
9. Finite Difference Schemes for Parabolic PDEs: 1D heat equation using FTCS, BTCS, and Crank-Nicolson schemes, compare their stability and accuracy.
10. Finite Difference Schemes for Hyperbolic PDEs: linear advection equation using upwind scheme, Lax-Wendroff scheme.

**Text Books:**

1. Kincaid and W. Cheney, Numerical Analysis: Mathematics of Scientific Computing, 3rd Edition, AMS, 2002.

2. D. Smith, Numerical Solutions of Partial Differential Equations, 3rd Edition, Calrendorn Press, 1985.

**Reference Books:**

1. E. Atkinson, An Introduction to Numerical Analysis, Wiley, 1989.
2. D. Conte and C. de Boor, Elementary Numerical Analysis - An Algorithmic Approach, McGraw-Hill, 1981.
3. Mitchell and S. D. F. Griffiths, The Finite Difference Methods in Partial Differential Equations, Wiley, 1980.
4. L. Burden and J. D. Faires, Numerical Analysis, Brooks/Cole, 2001.

Course Code	Course Name	L-T-P: C
HS202	Economics	3-0-0: 3

### Course Description

This course introduces B.Tech. students to the fundamental principles of economics and their applications in technology-driven environments. It covers microeconomic foundations (demand, supply, production, cost, and market structures), macroeconomic frameworks (national income, aggregate demand-supply, monetary and fiscal policy), and international trade. Two newly added modules broaden the intellectual scope of the course: the evolution of Economic Thought from Classical to Behavioural Economics, and the structure and policy landscape of the Indian Economy. Applications are drawn from the Indian technology industry, digital platforms, start-up ecosystems, and public policy, making economics immediately relevant to engineering graduates.

**Objectives**

1. Develop an understanding of core microeconomic and macroeconomic principles and their application to technology-intensive industries.
2. Acquaint students with the historical development of economic thought and the contribution of major economists.
3. Enable students to analyse the structure, challenges, and policy environment of the Indian economy.
4. Foster critical evaluation of market failures, government intervention, and economic policy trade-offs.
5. Build quantitative and analytical skills relevant to business decision-making, entrepreneurship, and public policy.

## Learning Outcomes

On successful completion of this course, students should be able to:

1. Apply demand-supply analysis and elasticity concepts to pricing and resource allocation decisions in technology markets.
2. Analyse production and cost structures relevant to software, hardware, and platform industries.
3. Evaluate market structures (perfect competition, monopoly, oligopoly) and their implications for tech firm strategy.
4. Interpret macroeconomic indicators — GDP, inflation, interest rates — and their impact on business and investment decisions.
5. Explain the evolution of economic thought and relate classical, Keynesian, monetarist, and behavioural frameworks to contemporary issues.
6. Assess the structure of the Indian economy, its growth trajectory, and the role of technology-led reforms in development.
7. Analyse international trade patterns and balance of payments with reference to India's technology services sector.

**Prerequisite:** MA101, MA201

### Module I: Foundations of Economic Organization and Technology

1. Problems of Economic Organization: Scarcity, choice, and opportunity cost, Economic systems: Market, Command, Mixed economies, Role of technology in addressing economic problems. Application: Resource allocation in cloud computing, optimization algorithms.
2. Demand, Supply, and Price Determination: Law of demand and supply, Market equilibrium and shifts, Price mechanism and market signals. Application: Dynamic pricing in e-commerce, surge pricing (Uber, Ola).
3. Elasticity Analysis: Price elasticity of demand and supply, Income and cross-price elasticity. Applications in business decisions. Application: Demand forecasting using data analytics, pricing strategies for software products (SaaS models)

### Module II: Production Economics and Technology

1. Theory of Production: Production function and technological relationships, Short-run vs. long-run production, Law of diminishing returns and its implications. Application: Moore's Law, returns to scale in manufacturing sector.
2. Production Optimization: Isoquants and iso-costs, Optimal input combination, Technical efficiency and productivity. Application: Manufacturing optimization, automation trade-offs, Industry 4.0.

3. Cost Analysis: Fixed, variable, and marginal costs, Short-run and long-run cost curves, Economies and diseconomies of scale. Tech Application: Cost structures in software vs. hardware industries, marginal cost pricing in digital goods.

### Module III: Forms of Market

1. Types of Market: Characteristics and assumptions, Price determination and firm behaviour, Short-run and long-run equilibrium and its application.
2. Platform Economics and Network Effects: Two-sided markets and platform business models, Network externalities and winner-take-all dynamics, Data as an economic resource. Application: Google, Amazon, Facebook, Airbnb, Uber business models

### Module IV: Macroeconomics Fundamentals

1. National Income Accounting: GDP, GNP, NNP concepts, Nominal vs. Real GDP, Methods of calculation.
2. Aggregate Demand and Supply Model: Components of aggregate demand, Aggregate supply in short and long run, Macroeconomic equilibrium. Application: Impact of technology on productivity and aggregate supply.
3. Consumption, Saving, and Investment: Keynesian consumption function, Marginal propensity to consume and save, Investment determinants and multiplier effect.

### Module V: International Trade and Cycle

1. International Trade Theories and Free Trade vs Protectionism, Comparative advantage and Heckscher-Ohlin theory, New trade theory and economies of scale, Free trade agreements and WTO framework, Arguments for protectionism: infant industry, strategic trade policy.
2. Balance of Payments: Current and capital account, Exchange rate determination, BOP disequilibrium and adjustment. Application: Outsourcing, tech services exports from India

### Module VI: Money, Banking and Finance

1. Money Market and Interest Rates: Demand for money (Liquidity preference theory), Interest rate determination, Term structure of interest rates.
2. Central Banking and Monetary Policy: Functions of Reserve Bank of India, Tools of monetary policy (CRR, SLR, Repo Rate, Open Market Operations), Monetary transmission mechanism.
3. Inflation Economics: Definition and measurement (CPI, WPI), Types of inflation (demand-pull, cost-push), Causes and consequences, Index numbers and their construction

**Module VII: Economic Thoughts and Schools of Economic Theory**

1. Pre-Classical and Classical Economics: Mercantilism: Trade surpluses and bullionism, Physiocrats: Land as the source of wealth (Quesnay), Adam Smith: Wealth of Nations, invisible hand, division of labour, David Ricardo: Comparative advantage, theory of rent, Thomas Malthus: Population theory and its critique, John Stuart Mill: Utilitarianism and economics.
2. Marxist Economics: Labour theory of value, Surplus value and capital accumulation, Critique of capitalism and historical materialism, Relevance to debates on automation and labour displacement.
3. Neoclassical Economics: Marginalism: Jevons, Menger, Walras, Utility theory and consumer equilibrium, Partial equilibrium (Alfred Marshall), General equilibrium (Leon Walras).
4. Keynesian Revolution: The General Theory: effective demand and involuntary unemployment, Fiscal policy as economic stabilizer, Multiplier and accelerator, Post-Keynesian and New Keynesian developments.
5. Monetarism and New Classical Economics: Milton Friedman: Quantity theory of money and natural rate of unemployment, Rational expectations hypothesis (Lucas critique), Real Business Cycle theory, Critique of activist fiscal policy.
6. Institutional and Behavioural Economics: Institutional economics: Veblen, Galbraith, Bounded rationality (Herbert Simon), Prospect theory and loss aversion (Kahneman & Tversky), Nudge theory and policy applications (Thaler & Sunstein). Application: Behavioural insights in digital product design, fintech.
7. Development and Welfare Economics: Sen's capability approach and human development, Amartya Sen: Social choice theory and welfare economics, Dependency theory and structuralism, Pareto efficiency vs. distributive justice.

**Module VIII: Indian Economy — Structure, Challenges, and Policy**

1. Overview and Historical Context: Colonial economy: Drain theory, de-industrialisation, Post-independence planning: Five-Year Plans and mixed economy model, License Raj and its consequences, Economic reforms of 1991: Liberalisation, Privatisation, Globalisation (LPG).
2. Structure of the Indian Economy: Sectoral composition: Agriculture, Industry, Services, Contribution of IT-BPM sector to GDP and exports, Informal economy: size, significance, and challenges, Labour market: Employment trends, demographic dividend, gig economy.

3. Agriculture and Rural Economy: Role of agriculture in India's GDP and employment, Green Revolution and its legacy, Agricultural reforms: MSP, APMC, contract farming debates, Technology in agriculture: precision farming, agri-tech start-ups.
4. Industry and Infrastructure: Industrial policy: Make in India, PLI schemes, Atma-nirbhar Bharat, Public sector enterprises vs. private sector growth, Infrastructure financing: PPP model, National Infrastructure Pipeline, Special Economic Zones (SEZs) and GIFT City.
5. Public Finance and Fiscal Policy in India: Union Budget: structure, revenue vs. capital expenditure, GST: design, implementation, and impact, Fiscal deficit, FRBM Act, and debt sustainability, Centre-State fiscal relations and Finance Commission.
6. Poverty, Inequality, and Human Development: Poverty measurement: Tendulkar, Rangarajan committees, Social protection: MGNREGS, PM-KISAN, Direct Benefit Transfer, Human Development Index: India's performance and gaps, Digital divide and inclusive growth.
7. Digital Economy and Technology-Led Growth: India Stack: Aadhaar, UPI, DigiLocker — economic impact, Start-up India and innovation ecosystem, Fintech revolution: digital payments, neobanks, BNPL, Artificial Intelligence, Industry 4.0, and the future of work in India, Data economy: personal data protection and economic value.
8. External Sector and Trade Policy: India's trade profile: major exports (IT services, pharmaceuticals, gems) and imports, Current account deficit and management, Foreign Direct Investment: trends, policy, and sectoral caps, India and multilateral trade: WTO disputes, bilateral FTAs.

**Text Books:**

1. Samuelson, P. A., & Nordhaus, W. D. (2010). Economics (19th ed.). McGraw-Hill Education.
2. Mankiw, N. G. (2021). Principles of Economics (9th ed.). Cengage Learning.
3. Pindyck, R. S., & Rubinfeld, D. L. (2018). Microeconomics (9th ed.). Pearson Education.
4. Dornbusch, R., Fischer, S., & Startz, R. (2018). Macroeconomics (13th ed.). McGraw-Hill Education.
5. Misra, S. K., & Puri, V. K. Indian Economy (43rd ed.). Himalaya Publishing House.
6. Datt, G., & Mahajan, A. Indian Economy (73rd ed.). S. Chand & Company.
7. Screpanti, E., & Zamagni, S. (2005). An Outline of the History of Economic Thought (2nd ed.). Oxford University Press.

- Thaler, R. H., & Sunstein, C. R. (2008). *Nudge: Improving Decisions about Health, Wealth, and Happiness*. Yale University Press.

#### Reference Books:

- Varian, H. R. (2014). *Intermediate Microeconomics: A Modern Approach* (9th ed.). W. W. Norton & Company.
- Blanchard, O. (2021). *Macroeconomics* (8th ed.). Pearson Education.
- Krugman, P., & Wells, R. (2021). *Economics* (6th ed.). Worth Publishers.
- Sen, A. (1999). *Development as Freedom*. Oxford University Press.
- Kahneman, D. (2011). *Thinking, Fast and Slow*. Farrar, Straus and Giroux.
- Reserve Bank of India. (Annual). *Annual Report*. Reserve Bank of India. [Available at [www.rbi.org.in](http://www.rbi.org.in)]
- Government of India. (Annual). *Economic Survey*. Ministry of Finance. [Available at [www.indiabudget.gov.in](http://www.indiabudget.gov.in)]
- Basu, K. (2004). *India's Emerging Economy: Performance and Prospects in the 1990s and Beyond*. MIT Press.
- Tirole, J. (2017). *Economics for the Common Good*. Princeton University Press.
- Parker, G. G., Van Alstyne, M. W., & Choudary, S. P. (2016). *Platform Revolution: How Networked Markets Are Transforming the Economy*. W. W. Norton & Company.

Course Code	Course Name	L-T-P: C
CS204	Database Management System	3-0-0: 3

**Objectives:** The course aims the students will be able to list and understand the basic concepts of a relational database system. They can analyze database requirements, determine the entities involved and relationships among them. The students are able to efficiently and effectively organize, maintain and retrieve information from a database system.

**Learning Outcomes:** On completion of this course the student should be able to:

- List the basic concepts of relational database model, relational algebra, entity-relationship model, and SQL.
- Convert the entity-relationship model to relational database and formulate SQL queries on it.
- Optimize database design through different processes.

#### Contents:

**Module I:** Introduction and Conceptual Modeling: Databases and database users; database system concepts and architecture; data modeling using the entity-relationship (ER) model; enhanced entity relationship.

**Module II:** Data Storage and Indexing: Introduction, record storage, and primary file organization index structures for files; single level indexing; multilevel indexing.

**Module III:** Relational Model: The relational data model; relational database constraints; relational algebra; relational calculus; relational database design by ER and EER; relational mapping; SQL; the relational database standard; examples of relational database management systems; Oracle.

**Module IV:** Database Design Theory and Methodology: Functional dependencies and normalization for relational databases, relational database design algorithms and further dependencies.

**Module V:** System Implementation Techniques: Query processing and optimization, transaction processing concepts, concurrency control techniques, database recovery techniques.

**Module VI:** Emerging Applications: Distributed databases and client/server models, XML Database (DTD, XML Schema), Query for XML Database, NoSQL.

#### Text Book:

- Fundamentals of Database Systems*, R. Elmasri, S. B. Navathe, Prentice Hall, New Delhi, 2007.

#### Reference Books:

- Database System Concepts*, A. Silberschatz, H. F. Korth, S. Sudharshan, Tata McGraw Hill, New Delhi, 2005.
- Introduction to Database Systems*, C. J. Date, Prentice Hall, New Delhi, 2004.

Course Code	Course Name	L-T-P: C
CS264	Database Management System Lab	0-0-2: 1

**Objectives:** To provide practical exposure to database design, SQL, and transaction management concepts.

**Learning Outcomes:** Students will be able to design databases, write SQL queries, and manage data using DBMS tools effectively.

#### Tentative List of Lab Programs

**Lab Program 1** involves designing an ER diagram for a real-world application such as a library or hospital, identifying entities, attributes, relationships, and cardinalities.

**Lab Program 2** focuses on converting ER diagrams into Enhanced ER (EER) diagrams and then mapping them to relational schemas.

**Lab Program 3** includes creating tables with constraints in SQL and performing basic operations like inserting, updating, deleting, and querying data using SELECT statements with WHERE, GROUP BY, and ORDER BY clauses.

**Lab Program 4** requires students to write advanced SQL queries using joins (inner, outer), nested queries, and set operations to retrieve data from related tables.

**Lab Program 5** introduces file organization and indexing concepts, where students simulate primary file organizations and create indexes to demonstrate single-level and multilevel indexing.

**Lab Program 6** covers relational algebra and calculus, requiring students to perform operations such as selection, projection, and joins manually, along with writing relational calculus expressions.

**Lab Program 7** focuses on database normalization, identifying functional dependencies, and applying 1NF, 2NF, 3NF, and BCNF to optimize table designs.

**Lab Program 8** introduces transaction management using SQL commands like BEGIN, COMMIT, and ROLLBACK, and demonstrates concurrency control using locking and isolation levels.

**Lab Program 9** allows students to analyze query performance using EXPLAIN plans in SQL databases, exploring different query strategies and indexing effects.

**Lab Program 10** involves working with NoSQL databases such as MongoDB for document storage and creating XML documents with DTD or XML Schema and querying them using XPath or XQuery.

Course Code	Course Name	L-T-P: C
AI206	Computer Organization and Architecture	3-0-0:3

**Objectives:** To study the fundamentals of digital logic design and computer architecture of digital computers (CPU, memory, I/O, software, pipelining and parallelism). Discussions will include digital logic, microprogramming and performance enhancement of

processors. Such knowledge leads to better understanding computer organization and architecture, can be used in the design and implementation of computer systems or as foundation for more advanced computer-related studies.

**Learning Outcomes:**

1. The students will be able to understand the fundamentals of digital logic design and circuits.
2. The students will be able to understand the design and implementation of 4. digital circuits.
3. To provide hands-on experience of software and hardware.
4. To familiarize the students with the processorinteraction with other hardware and time management.
5. To familiarize the students with pipeline architecture and instruction level parallelism.
6. To familiarize the students with dynamic instruction scheduling and thread level parallelism.
7. To familiarize the students with the memory system and its interaction with the processor.

**Contents:**

**Module I:** Number systems and Boolean algebra: Different number systems, conversion between them, binary arithmetic, complement representation, floating-point representation, logic gates, Boolean algebra laws, De Morgan’s theorem, Sum of Products (SOP) and Product of Sums (POS).

**Module II:** Combinational and Sequential logic: logic minimization using K-map method, QuineMcCluskey method, multiplexers/ demultiplexers, decoders/encoders, adders, ripple carry adder, carry-lookahead adder, multiplication - shift-and-add, Booth multiplier. Simple circuits with feedback, basic latches, clocks, Flip flops, shift register, synchronous and asynchronous counters, Finite State Machine (Moore/Mealy Machines).

**Module III:** Von Neumann: Functional units, ALU, data paths architecture, registers, instruction set architecture (ISA), addressing modes.

**Module IV:** Pipelining and Parallelism: An overview of pipelining, throughput and speedup, pipelined data path and control, data dependency and hazard, control hazard and structural hazard, instruction level parallelism (ILP) concepts and challenges, basic compiler techniques for exposing ILP, ILP using dynamic scheduling, VLIW, superscalar architecture, overview of thread level parallelism. Introduction to data-level parallelism (DLP) and comparison with ILP and TLP; SIMD and SIMT execution models; basic GPU architecture overview including thread hierarchy and execution model; limitations of ILP in modern

workloads and motivation for throughput-oriented architectures.

**Module V:** Memory system and I/O: Principles of temporal and spatial locality; cache memories (address mapping, block size, replacement and store policy); virtual memory (page table, TLB); disk organization and data access from disk drive, programmed I/O, interrupt-driven I/O, Handshaking, DMA, interrupts. Memory hierarchy considerations in parallel architectures, including basic GPU memory organization and the impact of memory access patterns on performance and bandwidth utilization.

**Text Books:**

1. Digital Design, Morris Mano, Prentice Hall, 2002.
2. Digital Fundamentals, 10th Ed, Floyd T L, Prentice Hall, 2009.
3. Computer Architecture: A Quantitative Approach, 5th Edition By John L Hennessy and David A Patterson.

**Reference Books:**

1. Digital electronics: principles, devices and applications, Albert Paul Malvino and Jerald A. Brown, John Wiley & Sons, 2007.
2. Computer system architecture, Morris Mano, revised third edition, Pearson.
3. Computer Organization and Architecture, 8th Ed, Stallings, Pearson, 2010.

Course Code	Course Name	L-T-P: C
AI266	Computer Organization and Architecture Lab	0-0-2:1

**Objectives:** To provide fundamental understanding of computer organization and architecture, including data representation, processor design, memory hierarchy, and input-output systems.

**Learning Outcomes:** Students will be able to analyze computer system components, understand instruction execution and processor design, evaluate memory and I/O mechanisms, and apply concepts for efficient system performance.

**Tentative List of Lab Experiments:**

1. Design and implementation of data path architecture and program counter.
2. Design and implementation of arithmetic logic unit (ALU).
3. Design and implementation of ripple carry adder and carry-look-ahead adder.

4. Design and implementation of Booth's multiplier and combinational multipliers.
5. To understand the basic principles of pipeline design for single instruction, including the problems of data and branch hazards.
6. To understand pipeline design for multiple instructions, including the problems of data and branch hazards.
7. To understand the multiple instructions (beq, lw and st) working using pipelined processors.
8. To understand the arithmetic instruction (add, mult, div, sub) working using instruction level parallelism.
9. To understand the implementation of instruction level parallelism using ScoreBoard algorithm.
10. To understand the implementation of instruction level parallelism using the Tomasulo algorithm.
11. To understand Data-Level Parallelism (DLP) by comparing scalar and SIMD-style execution.
12. To understand how memory access patterns affect performance, especially in parallel systems.

**Text Books:**

1. Digital Fundamentals, 10th Ed, Floyd T L, Prentice Hall, 2009.
2. Computer Architecture: A Quantitative Approach, 5th Edition By John L Hennessy; David A Patterson.

**Reference Books:**

1. Computer system architecture, Morris Mano, revised third edition, Pearson.
2. Computer Organization and Architecture, 8th Ed, Stallings, Pearson, 2010.

Course Code	Course Name	L-T-P: C
CS208	Computer Organization and Architecture	3-0-0: 3

**Objectives:** To study the fundamentals of digital logic design and computer architecture of digital computers (CPU, memory, I/O, software, pipelining and parallelism). Discussions will include digital logic, microprogramming and performance enhancement of processors. Such knowledge leads to better understanding computer organization and architecture, can be used in the design and implementation of computer systems or as foundation for more advanced computer-related studies.

**Learning Outcomes:**

1. The students will be able to understand the fundamentals of digital logic design and circuits.

2. The students will be able to understand the design and implementation of 4. digital circuits.
3. To provide hands-on experience of software and hardware.
4. To familiarize the students with the processor interaction with other hardware and time management.
5. To familiarize the students with pipeline architecture and instruction level parallelism.
6. To familiarize the students with dynamic instruction scheduling and thread level parallelism.
7. To familiarize the students with the memory system and its interaction with the processor.

**Contents:**

**Module I:** Number systems and Boolean algebra: Different number systems, conversion between them, binary arithmetic, complement representation, floating-point representation, logic gates, Boolean algebra laws, De Morgan’s theorem, Sum of Products (SOP) and Product of Sums (POS).

**Module II:** Combinational and Sequential logic: logic minimization using K-map method, QuineMcCluskey method, multiplexers/ demultiplexers, decoders/encoders, adders, ripple carry adder, carry-lookahead adder, multiplication - shift-and-add, Booth multiplier. Simple circuits with feedback, basic latches, clocks, Flip flops, shift register, synchronous and asynchronous counters, Finite State Machine (Moore/Mealy Machines).

**Module III:** Von Neumann: Functional units, ALU, data paths architecture, registers, instruction set architecture (ISA), addressing modes.

**Module IV:** Pipelining and Parallelism: An overview of pipelining, throughput and speedup, pipelined data path and control, data dependency and hazard, control hazard and structural hazard, instruction level parallelism (ILP) concepts and challenges, basic compiler techniques for exposing ILP, ILP using dynamic scheduling, VLIW, superscalar architecture, overview of thread level parallelism.

**Module V:** Memory system and I/O: Principles of temporal and spatial locality; cache memories (address mapping, block size, replacement and store policy); virtual memory (page table, TLB); disk organization and data access from disk drive, programmed I/O, interrupt- driven I/O, Handshaking, DMA, interrupts.

**Text Books:**

1. Digital Design, Morris Mano, Prentice Hall, 2002.
2. Digital Fundamentals, 10 th Ed, Floyd T L, Prentice Hall, 2009.

3. Computer Architecture: A Quantitative Approach, 5th Edition By John L Hennessy and David A Patterson.

**Reference Books:**

1. Digital electronics: principles, devices and applications, Albert Paul Malvino and Jerald A. Brown, John Wiley & Sons, 2007.
2. Computer system architecture, Morris Mano, revised third edition, Pearson.
3. Computer Organization and Architecture, 8th Ed, Stallings, Pearson, 2010.

Course Code	Course Name	L-T-P: C
CS268	Computer Organization and Architecture Lab	0-0-2: 1

**Objectives:** To provide fundamental understanding of computer organization and architecture, including data representation, processor design, memory hierarchy, and input-output systems.

**Learning Outcomes:** Students will be able to analyze computer system components, understand instruction execution and processor design, evaluate memory and I/O mechanisms, and apply concepts for efficient system performance.

**Tentative List of Lab Experiments:**

1. Design and implementation of data path architecture and program counter.
2. Design and implementation of arithmetic logic unit (ALU).
3. Design and implementation of ripple carry adder and carry-look-ahead adder.
4. Design and implementation of Booth’s multiplier and combinational multipliers.
5. To understand the basic principles of pipeline design for single instruction, including the problems of data and branch hazards.
6. To understand pipeline design for multiple instructions, including the problems of data and branch hazards.
7. To understand the multiple instructions (beq, lw and st) working using pipelined processors.
8. To understand the arithmetic instruction (add, mult, div, sub) working using instruction level parallelism.
9. To understand the implementation of instruction level parallelism using ScoreBoard algorithm.
10. To understand the implementation of instruction level parallelism using the Tomasulo algorithm.

**Text Books:**

1. Digital Fundamentals, 10th Ed, Floyd T L, Prentice Hall, 2009.
2. Computer Architecture: A Quantitative Approach, 5th Edition By John L Hennessy; David A Patterson.

#### Reference Books:

1. Computer system architecture, Morris Mano, revised third edition, Pearson.
2. Computer Organization and Architecture, 8th Ed, Stallings, Pearson, 2010.

Course Code	Course Name	L-T-P: C
PH202	Foundations of Quantum Technologies	3-0-0-3

**Objectives:** The course presents the central theoretical aspects of quantum mechanics in a rigorous manner, where students learn the techniques and develop a solid intuition for quantum physics.

**Learning Outcomes:** Upon successful completion of the course, students will be able to

1. Understand the most relevant mathematical techniques.
2. Understand the basic postulates of quantum mechanics and applications.
3. Understand the basics of Statistical Physics.
4. Understand the basics of Information Science.
5. Understand the basics of computational complexity.

**Prerequisites:** PH101, PH104

#### Contents:

**Module-I:** Postulates of Quantum Mechanics: State vectors and Hilbert Space, Dirac Bra-Ket notation, Measurables and Hermitian Operators, Unitary Transformations, Schrodinger Equation and Time evolution of quantum states, Measurement Postulate, Schrodinger, Heisenberg and Interaction pictures, Eigen values, Expectation values and Matrix elements, Heisenberg Uncertainty principle.

**Module-II:** Applications of Postulates: Particle in a box, Hydrogen atom, Harmonic Oscillator, Number states, ladder operators and Coherent states of a harmonic oscillator, Spin and Angular momentum – spin half particles, Rabi problem of a spin-half particle in a rotating magnetic field, Bosons and Fermions.

**Module-III:** Crystal structure, Bravais lattices, Unit cell, Miller indices, Symmetry, Brillouin zones, Reciprocal Lattice, X-ray diffraction and Bragg's law, Atomic scattering and structure factors, Defects and dislocations.

**Module-IV:** Free electron theory of metals: Drude and Sommerfeld models, density of states, Fermi–Dirac distribution, Fermi energy, Fermi surface, Electrons in periodic potentials: Bloch theorem, Kronig–Penney model, Energy bands, direct and indirect band gaps, effective mass and holes, Electrical conductivity of metals, scattering mechanisms and low-temperature transport, Semiconductors: intrinsic and extrinsic, carrier concentrations, effective density of states, electrical conductivity and mobility, temperature dependence, and Hall effect.

#### Text/Reference Books:

1. Introduction to Quantum Mechanics, Griffiths D. J., 3rd Edition, Cambridge University Press (2024).
2. Principles of Quantum Mechanics, Shankar, R., 2nd edition, Springer (2014).
3. C. Kittel, Introduction to Solids State Physics.
4. N. W. Ashcroft and N. D. Mermin, Solids State Physics

Course Code	Course Name	L-T-P: C
PH262	Laboratory for Quantum Technologies	0-0-2-1

**Objectives:** The course aims to provide hands-on experience in applying quantum mechanics principles to physical systems. It focuses on developing practical skills in quantum optics, circuits, sensing, and control, preparing students to design, analyze, and build hardware for quantum computing and communication.

**Learning Outcomes:** Upon successful completion of the course, students will be able to

1. Perform basic experiments in optics.
2. Perform basic experiments in characterising resonators and RLC circuits.
3. Design basic digital circuits.
4. Understand fundamental techniques in RF engineering.
5. Interface instruments with computers and carry out data acquisition.

**Prerequisites:** PH161, EC161, EC162, EC261

#### Contents:

**Experimental Module-I:** Optics: Interferometry – wavelength measurements, intensity measurements, Diffraction – single slit, grating, Microscopy – magnification, aberration, Polarization optics – PBS, HWP, QWP

**Experimental Module-II:** Circuits: RLC circuits: Series and parallel RLC circuits – Verifying the quality factor formulae, Extracting intrinsic losses. Digital

circuits: Adder, Multiplier, Encoder, Decoder, D flipflop, shift registers, How to use common Integrated Circuit chips

**Experimental Module-III:** Radio Frequency Technology: Using Oscilloscope: Ring-up and ring-down time measurements of RLC circuits, Measurements of different pulse-shapes generated by a function generator. Using Vector Network Analyser: Transmission and reflection measurements of coaxial cable in open, short and matched termination, Voltage standing wave ratio measurement, Amplitude and Phase quadrature, In-phase and Out-of-phase quadrature plots and Quality factor measurement of RLC circuits, Characterising S-parameters, ABCD and Z matrices of common 2 port networks – coaxial cable, attenuator, low pass high pass bandpass filters etc.Characterising 3 port networks – directional couplers, circulators, isolators. Using a Spectrum Analyser: Noise from a resistor at different temperatures

**Experimental Module-IV:** Interfacing instruments with a computer. Data acquisition: Signal demodulation – heterodyne vs Homodyne, Mixing of signals, Sampling, digitisation using ADCs – under-sampling and aliasing, oversampling and noise, Averaging and interpolation techniques, Quantum Simulators, Running quantum protocols in a quantum simulator, Implementing simple quantum algorithms on cloud-based quantum computers (depending on availability of time on such machines), Running simple algorithms on cloud-based quantum processors (optional)

**Text/Reference Books:**

1. Optics, Eugene Hecht, A. R. Ganesan, 5th edition, Pearson (2019).
2. Art of Electronics, Paul Horowitz and Winfield Hill, 3rd edition, Cambridge University Press (2015).
3. Digital Design, Morris Mano, Michael D. Ciletti, 6th edition, Pearson Education (2018).
4. Microwave Engineering, David Pozar, 4th edition, Wiley (2013).
5. Discrete-time signal processing, Alan V. Oppenheim and Ronald W. Shaffer, 4th edition, Pearson (2009).
6. Optical quantum information and quantum communication, A. Pathak and A. Banerjee, SPIE Spotlight Series, SPIE Press (2016)

Course Code	Course Name	L-T-P: C
AI202/ MC206	Machine Learning	3-0-2:4

**Objectives:** The course provides a comprehensive introduction to core Machine Learning (ML) methods, covering both supervised and unsupervised learning, and their applications in AI and data analysis. Students will be enabled to formulate ML problems and implement key algorithms, such as linear and logistic regression, including model training and evaluation. A strong emphasis is placed on the vital role of ML in modern AI, recognizing it as the engine; powering intelligent systems, while also addressing the growing industry demand for ML expertise.

**Learning Outcomes:**

1. Understand foundational algorithms: Explain and implement core supervised methods (linear regression, logistic regression).
2. Model training and evaluation: Apply optimization (gradient descent) to train models and use metrics (e.g., MSE, accuracy) to assess performance.
3. Apply unsupervised techniques: Use clustering (k-means) and dimensionality- reduction (PCA) methods to explore and compress data.
4. Deep learning basics: Describe artificial neural networks (perceptrons, multilayer perceptrons) and key deep models (CNNs, autoencoders), understanding their structure and uses. Prerequisite: Linear algebra, probability and statistics, and calculus

**Contents:**

**Module 1:** Supervised Learning and Linear Models: Introduction to Machine Learning: Definitions and types (supervised vs. unsupervised learning) ; ML workflow (data, training, testing). Linear Regression: Model formulation, cost functions (MSE), normal equations and gradient descent methods; bias-variance tradeoff and regularization (L2, L1). Logistic Regression: Binary classification using the sigmoid function; cost/loss formulation (log loss), decision boundary; extending to multiclass. Performance Evaluation: Loss functions, cross-validation, overfitting vs. underfitting.

**Module 2:** Advanced Supervised Learning: Linear Classifiers and Perceptrons: Perceptron learning algorithm and its convergence; relationship to logistic regression and SVM. Support Vector Machines (SVM): Maximum-margin classification; soft margins and hinge loss. Kernel Methods: Mapping to high-dimensional spaces using kernel functions (linear, polynomial, RBF); the kernel trick for non-linear decision boundaries. Probabilistic Models: Brief introduction to generative models (Gaussian Discriminant Analysis, Naïve Bayes) and their assumptions.

**Module 3:** Unsupervised Learning and Dimensionality Reduction Clustering: K-means algorithm, initialization and convergence; hierarchical and density- based clustering. Mixture Models and EM: Gaussian Mixture Models and the Expectation Maximization algorithm for parameter estimation. Principal Component Analysis (PCA): Theory of PCA for dimensionality reduction; computing principal components and data reconstruction.

**Module 4:** Neural Networks and Deep Learning: Artificial Neural Networks (ANNs): Perceptron vs. multi-layer perceptron (MLP); network architecture, activation functions. Training Neural Networks: Backpropagation algorithm and gradient-based optimization; issues of convergence and regularization. Convolutional Neural Networks (CNNs): Convolutional layers, pooling, and architecture for image data (basic intuition, no math depth needed at this level). Autoencoders: Concepts of encoding/decoding networks; sparse and denoising autoencoders; introduction to variational autoencoders for representation learning. Generative Adversarial Networks (GANs): High-level overview of GAN structure (generator vs. discriminator) and applications.

#### Text Books:

1. Tom M. Mitchell, Machine Learning (McGraw-Hill).
2. Christopher M. Bishop, Pattern Recognition and Machine Learning (Springer).
3. Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning (MIT Press).
4. Aurélien Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow (O'Reilly).

#### Reference Books:

1. Trevor Hastie, Robert Tibshirani, Jerome Friedman, The Elements of Statistical Learning (Springer).
2. Kevin P. Murphy, Machine Learning: A Probabilistic Perspective (MIT Press).
3. Yaser S. Abu-Mostafa, Malik Magdon-Ismael, Hsuan-Tien Lin, Learning from Data (AMLBook.com).
4. David Barber, Bayesian Reasoning and Machine Learning (Cambridge).

#### Lab Experiments:

1. Implement linear regression using gradient descent and analyze error convergence.
2. Apply logistic regression for binary classification and evaluate model performance.
3. Implement perceptron learning algorithm and compare with logistic regression.
4. Perform classification using Naïve Bayes and analyze the effect of Laplace smoothing.

5. Implement Support Vector Machine (SVM) with linear and kernel functions.
6. Perform clustering using K-means and analyze cluster quality.
7. Apply Principal Component Analysis (PCA) for dimensionality reduction and visualization.
8. Train a simple neural network using backpropagation for classification task.

Course Code	Course Name	L-T-P: C
AI204	Artificial Intelligence	3-0-0:3

**Description:** The objective of the course is to establish the fundamental theme of using computers to solve complex problems, beginning with the concept of intelligence as problem-solving ability. A core focus is on understanding difficult problems in computation, interpreting software as intelligent agents, and crucially, modeling these problems so they can be solved using computer programs to understand artificial intelligence. The ultimate goal is to build the necessary foundations for students to pursue advanced courses in the field of AI.

**Objectives:** The use of computers to solve complex problems is the fundamental theme of this course. The notion of intelligence captured by problem solving ability is a starting point of the course. Understanding the difficult problems in computation and interpreting software as intelligent agents is important. Modeling the problems in a way that can be solved using computer programs is very crucial to understanding artificial intelligence. The course aims at building the foundations for advanced courses in the field of AI.

**Learning Outcomes:** On completion of this course the student should be able to:

1. Students will be able to model a real world problem in a formal way with appropriate specifications, for example: search problems, multiplayer games and strategies, expert systems etc.
2. Student will be able to select suitable search strategy for the given problem.
3. Student will be able to setup a deep learning computational pipeline for solving moderate size classification and regression problems.
4. Student will be able to program computer to solve logical inference problems with uncertainty

**Prerequisites:** MA201, CS102, Design and Analysis of Algorithms

**Contents:**

**Module I: Introduction to AI and Search:** History, The state of the art; intelligent agents; structure; environment; Configuration and Planning Problems, State space representation, Breadth-first search; uniform cost search; depth-first search; depth-limited search; iterative, deepening search; bi-directional search; heuristic search techniques; comparing search strategies, randomized search, adversarial search, alpha-beta pruning

**Module II: Probabilistic Reasoning:** Probability and Bayes theorem; certainty factors; Bayesian networks (Graphical Models), Markov Networks; Hidden Markov Model, Decision Tree and Random Forest

**Module III: Neural Networks:** Neuron as a nonlinear combiner, Capacity of a single neuron, Feed-forward networks and Universal Approximation Theorem, Hopfield Network and Associative Memory

**Module IV: Reinforcement Learning and Computational Game Theory:** Markov Decision Process, Reinforcement Learning, Extensive form games, Social Choice and Mechanism Design

**Module V: Agentic AI Foundations:** Retrieval-augmented generation (RAG), Embeddings & vector search, Introduction to Multimodal pipelines (vision+text/audio integration patterns), Introduction to Multi-agent systems, Ethics, safety for deployed agents.

**Text Books:**

1. Artificial Intelligence – A Modern Approach, S. Russell, P. Norvig, Pearson Education, New Delhi, 2021 (Fourth Edition).
2. A First Course in Artificial Intelligence, Khemani D., Tata McGraw Hill, 2014.

**Reference Books:**

1. Reinforcement Learning: An Introduction, Second Edition, Sutton and Barto MIT Press, 2018.
2. Perceptrons - An Introduction to Computational Geometry, Marvin Minsky and Seymour Papert, The MIT Press, 1969.
3. Artificial Intelligence: A guide to intelligent systems, Negnevitsky M., Pearson Education, 2005.
4. The Quest for Artificial Intelligence: A history of ideas and achievements, Nilsson N., Cambridge University Press, 2008.
5. Selected Papers on Deep Belief Networks, Multi-agent Reinforcement Learning and Markov Logic Networks.

Course Code	Course Name	L-T-P: C
-------------	-------------	----------

AI264	Artificial Intelligence Lab	0-0-2:1
-------	-----------------------------	---------

**Description:** This course provides hands-on experience implementing foundational AI techniques including search, heuristic reasoning, probabilistic modeling, and reinforcement learning. Students will learn to implement and analyze search algorithms, develop AI models for uncertainty and sequential decision-making, and design intelligent agents.

**Objectives:** To provide hands-on experience in implementing foundational AI techniques including search, heuristic reasoning, probabilistic modeling, reinforcement learning, and sequential decision-making.

**Learning Outcomes:** On completion of this course the student should be able to:

1. Implement and experimentally analyze classical and heuristic search algorithms for solving structured AI problems.
2. Develop and evaluate AI models for uncertainty reasoning, reinforcement learning, and sequential decision making.
3. Design and implement intelligent agents integrating search, optimization, probabilistic inference, and learning techniques.

**Prerequisite:** MA201, CS102, Design and Analysis of Algorithms

**Tentative Laboratory Assignments:**

**Lab 1:** To be able to model a given problem in terms of state space search problem and solve the same using BFS/ DFS.

**Lab 2:** To design a graph search agent and understand the use of a hash table, queue in state space search.

**Lab 3:** To understand the use of Heuristic function for reducing the size of the search space. Explore non-classical search algorithms for large problems.

**Lab 4:** Non-deterministic Search | Simulated Annealing: For problems with large search spaces, randomized search becomes a meaningful option given partial/ full-information about the domain.

**Lab 5:** Understand the graphical models for inference under uncertainty, build Bayesian Network in R, Learn the structure and CPTs from Data, naive Bayes classification with dependency between features.

**Lab 6:** To understand the working of Hopfield network and use it for solving some interesting combinatorial problems.

**Lab 7:** Basics of data structure needed for state-space search tasks and use of random numbers required for MDP and RL, Understanding Exploitation -

Exploration in simple n-arm bandit reinforcement learning task, epsilon-greedy algorithm.

**Lab 8:** Understand the process of sequential decision making (stochastic environment) and the connection with reinforcement learning.

Course Code	Course Name	L-T-P: C
CS202	System Software	3-0-0: 3

**Objectives:** This course introduces design and implementation of various types of system software and their relationship with machine architecture.

**Learning Outcomes:** On successful completion of this course, students should be able to:

1. List relationship between machine architecture and system software.
2. Analyze different types of software processors viz. assemblers, compilers, loaders.
3. Able to differentiate between top down and bottom up parsing and understand syntax directed translation techniques.

**Contents:**

**Module I:** Introduction: Overview and history, Language Processors, Introduction to CISC and RISC machine architecture.

**Module II:** Assembler: Basic Assembler Functions, Machine Dependent Features, Machine Independent Features, One pass and Multi pass Assembler.

**Module III:** Linkers, Loaders and Macro Processors: Basic Loader Function, Loader Design Options, Relocation and Linking Concepts, Design of a Linker, Case study for Linker and Loader, Basic Macro Processor Functions and Features, Macro Processor Design Options, Implementation example for Macro Processor.

**Module IV:** Compilers: Aspects of Compilation, Compiler Features, Memory Allocation, Grammar, Parsing Techniques, Compiler Design Options, Intermediate Code Generation and Optimization Techniques.

**Module V: Software Tools:** Text Editors, Debuggers, Data Base Management System, User Interfaces.

**Text Books:**

1. *System Software – An introduction to System Programming*, Leland L. Beck, 3<sup>rd</sup> Edition, Pearson Education.

2. *Systems Programming and Operating Systems*, D. M. Dhamdhare, Tata McGraw Hill Publication.

**Reference Books:**

1. *System Programming*, John Donovan, McGraw Hill Publication.
2. *System Software*, Santanu Chattopadhyay, Prentice Hall India Publication.

Course Code	Course Name	L-T-P: C
CS206	Software Engineering	2-0-0: 2

**Objectives -** To introduce fundamental concepts of software engineering, including software processes, requirements analysis, design principles, and testing techniques for developing high-quality software systems.

**Learning Outcome -** Students will be able to analyze software requirements, design system models, apply testing techniques, and ensure software quality using standard engineering practices.

**Contents:**

**Module I:** The evolving role of software; software characteristics; software process - software process models; linear sequential model; prototyping model; The RAD model; evolutionary software process models; The incremental model; The spiral model.

**Module II:** System Engineering: Requirements analysis and negotiation; Requirements validation; Requirements management.

**Module III:** Analysis Modeling: Data modeling; data objects, attributes and relationships; cardinality and modality; entity-relationship diagram; data flow diagrams; data dictionary.

**Module IV:** Design Concepts and Principles: Software architecture; control hierarchy; structural partitioning; functional Independence; cohesion, coupling; design documentation; architectural design; transform centered architecture; transaction centered architecture; user Interface design models, user interface design process.

**Module V:** Testing Techniques: Software testing fundamentals; test case design; White box testing; basis path testing; control structure testing; Black box testing, testing for specialized environments, testing strategies; verification and validation - unit testing, integration testing, validation testing, system testing, debugging.

**Module VI:** Software Quality Assurance: Quality concepts; cost of quality, Software Quality Assurance (SQA) Group - roles and responsibilities, formal technical reviews, quality standards.

**Text Books:**

1. Software Engineering – A Practitioner’s Approach, R. S. Pressman, McGraw Hill International Edition, Singapore, 2006.

**Reference Books:**

1. Software Engineering, I. Sommerville, Pearson Education, New Delhi, 2001.
2. An Integrated Approach to Software Engineering, P. Jalote, Narosa Publishers, New Delhi, 2005.

Course Code	Course Name	L-T-P: C
CS266	Software Engineering Lab	0-1-2: 2

**Objectives:** To provide practical exposure to software engineering processes including requirements analysis, design modeling, testing, and quality assurance.

**Learning Outcomes:** Students will be able to develop requirement models, design software systems, apply testing techniques, and follow standard software engineering practices.

**Tentative List of Lab Exercises**

1. Preparation of Software Requirement Specification (SRS) for a given problem.
2. Development of use case diagrams and scenarios.
3. Design of Data Flow Diagrams (DFD).
4. Creation of Entity-Relationship (ER) diagrams.
5. Preparation of data dictionary and requirement models.
6. Design of software architecture and UML diagrams (class, sequence).
7. User Interface (UI) design for a sample application.
8. Writing test cases for given modules.
9. Implementation of white-box and black-box testing techniques.
10. Conducting software quality assurance activities and preparing test reports.

Course Code	Course Name	L-T-P: C
IT202	Web Technologies	3-0-0: 3

**Objectives:** This course introduces fundamental concepts of web development, covering HTML, CSS, and JavaScript for creating interactive web pages. It explores front-end and back-end technologies, including basic server-side scripting and databases. The course emphasizes hands-on practice in building responsive and dynamic websites.

**Learning Outcomes:** On successful completion of this course, students should be able to:

1. Gain knowledge of fundamental web technologies, including HTML, CSS, and JavaScript, for creating structured and styled web pages.
2. Learn to enhance user experience using JavaScript for dynamic and interactive web content.
3. Apply CSS frameworks and techniques to build mobile-friendly and visually appealing web pages.
4. Understand the basics of server-side scripting and database connectivity for dynamic web applications.

**Prerequisite:** None

**Contents:**

**Module I - Introduction to Web Programming:** Overview of the web and how it works, client-side vs. server-side programming, introduction to web browsers and developer tools

**Module II - HTML – Structuring Web Pages:** HTML syntax and document structure, HTML elements: headings, paragraphs, lists, tables, and forms, hyperlinks, images, and multimedia integration, semantic HTML and best practices

**Module III - CSS – Styling Web Pages:** Introduction to CSS and its role in web design, CSS selectors, properties, and units, box model, flexbox, and grid layout, responsive design with media queries

**Module IV – JavaScript: Basics of JavaScript:** variables, data types, and operators, functions, loops, and conditionals, DOM manipulation and event handling

**Module V - Web Forms and User Input:** Creating and styling forms, form validation using HTML5 and JavaScript, handling user input and feedback

**Module VI - Introduction to Frontend Frameworks:** Basics of Bootstrap or Tailwind CSS for faster development, introduction to JavaScript libraries (e.g., React)

**Module VII - Basics of Backend Development:** Understanding how the web server works, introduction to server-side programming (e.g., Node.js), basics of databases and SQL, sending and receiving data using HTTP requests

**Module VIII - Web Hosting and Deployment:** Introduction to web hosting services, deploying websites using GitHub Pages, Netlify, or FTP, basics of domain names and web hosting.

**Text Books:**

1. HTML and CSS: Design and Build Websites, Jon Duckett, Wiley (2011).

- Learning Web Design, Jennifer Robbins, O'Reilly (2018).

**Reference Books:**

- JavaScript: The Definitive Guide, David Flanagan, O'Reilly (2020).
- Web Development and Design Foundations with HTML5, Terry Felke-Morris, Pearson (2018).

Course Code	Course Name	L-T-P: C
IT262	Web Technology Lab	0-0-2: 1

**Objectives:** Students will apply web technology for building web applications. They will develop a case study where they will be able to understand and demonstrate use of web technology for a particular domain.

**Learning Objectives:** On successful completion of this course, students should be able to:

- Design and implement dynamic websites using appropriate and modern languages discussed.
- Create a client-server web application using appropriate tools.
- Connect a webpage to a database and perform relevant data access operations.
- Have a Good knowledge of web application terminologies, Internet tools, e-commerce and other web services.

**Prerequisites:** CS201

**Contents:**

Web technology tools, XML, SOAP, CORBA, RMI with emphasis on following:

**Project/Assignment-1 (Information Flow):**

Implementation of complete website; PHP backend; MySQL Database; front-end Form development (text, email, radio, checkbox, select/data list)

**Project/Assignment-2 (Validation and Structure):**

Client-side validation of project/assignment 1; Server-side validation of project/assignment 1; Object-oriented designing of PHP backend (following MVC architecture); Unit testing; Using Git; Using GitHub.

**Project/Assignment-3 (Session Management):**

Session Management addition to project/assignment 2; User login addition to project/assignment 2; Styling & Layout addition project/assignment 2.

Course Code	Course Name	L-T-P: C
CC202	Software and Hardware	3-0-2: 4

**Objectives:** To provide students with fundamental and advanced knowledge of software and hardware security, including vulnerabilities, threat models, secure design principles, and modern defense mechanisms.

**Learning Outcomes:** After completing this course, students will be able to:

- Understand common software and hardware vulnerabilities.
- Apply secure coding practices and threat modeling techniques.
- Analyze and mitigate system-level and hardware-based attacks.
- Evaluate security mechanisms in modern computing systems.
- Design secure systems integrating both software and hardware protections.

**Contents:**

**Module I:** Fundamentals of Security: Introduction to security principles: confidentiality, integrity, availability (CIA triad), threat models, attack surfaces, types of attackers, basics of cryptography, security policies and risk management.

**Module II:** Software Security: Secure software development lifecycle (SSDLC), common vulnerabilities (buffer overflow, SQL injection, XSS), secure coding practices, static and dynamic analysis tools, malware basics, sandboxing, and patch management. Memory corruption, Fuzzing, Penetration Testing: lifecycle, tools.

**Module III:** System and OS Security: Operating system security, access control models (DAC, MAC, RBAC), authentication mechanisms, file system security, virtualization security, container security, intrusion detection and prevention systems.

**Module IV:** Hardware Security and Emerging Trends: Introduction to hardware security, side-channel attacks, hardware Trojans, trusted platform module (TPM), secure boot, embedded system security, IoT hardware vulnerabilities, physical attacks and countermeasures. Cloud security, edge and IoT security, blockchain basics for security, AI in cybersecurity, privacy-preserving techniques, case studies and recent attack scenarios.

**Textbooks:**

- Computer Security: Principles and Practice, William Stallings and Lawrie Brown, Computer Security: Principles and Practice, 4th ed., Pearson, 2018.

2. Security Engineering, Ross Anderson, Security Engineering: A Guide to Building Dependable Distributed Systems, 3rd ed., Wiley, 2020.
3. The Hardware Hacker, Andrew Huang, The Hardware Hacker: Adventures in Making and Breaking Hardware, No Starch Press, 2017.

**Reference Books:**

1. Hacking: The Art of Exploitation, Jon Erickson, Hacking: The Art of Exploitation, 2nd ed., No Starch Press, 2008.
2. Applied Cryptography: Protocols, Algorithms, and Source Code in C, Bruce Schneier, 2nd ed., Wiley, 1996.
3. Introduction to Hardware Security and Trust, Mark Tehranipoor and Cliff Wang, Introduction to Hardware Security and Trust, Springer, 2011.

Course Code	Course Name	L-T-P: C
EC202	IC Design	3-1-0: 4

**Objectives:** A Digital IC Design course typically focuses on the principles and techniques used in designing integrated circuits (ICs) for digital systems. The course objectives include providing students with a strong understanding of the fundamental concepts of digital electronics, circuit design, and the operation of logic gates.

**Learning Outcomes:**

1. Understanding digital logic: Comprehend the fundamental concepts of digital logic, including Boolean algebra, logic gates, and combinational and sequential circuits.
2. Proficiency in design tools: Gain experience with industry-standard design tools, including Hardware Description Languages (HDLs) like VHDL/Verilog and simulation tools for circuit modeling and analysis.
3. Circuit optimization: Learn how to optimize digital circuits for performance, power, and area efficiency.
4. Timing and verification: Understand timing analysis, constraints, and verification techniques to ensure correct functionality and performance of digital designs.
5. Hands-on project experience: Apply theoretical knowledge to real-world projects, developing practical design solutions and performing tests and simulations.

6. Testing and debugging: Develop skills to test and debug digital circuits and troubleshoot design issues in a systematic manner.
7. Design flow understanding: Gain a comprehensive understanding of the entire digital design flow, from specification and design to verification and fabrication.

**Prerequisite:** Digital logic design, Electronic Devices and Circuits and SPICE

**Contents:**

**Module-I:** Silicon and Doping, P-N Junction, CMOS Transistor, Threshold Voltage, ON Current, Channel length modulation, Velocity saturation, Sub-threshold leakage, Drain Induced Barrier Leakage, Gate Induced Drain leakage, (Reverse) Short Channel Effect, Other leakage mechanisms, Capacitance Resistance

**Module-II:** Capacitance, Resistance, Sheet Resistance, Skin depth, Resistance Models, Lumped model (C and RC), Propagation delay and rise time, Elmore delay model, Example - Time constant of a re-wire model

**Module-III:** Switch Model, Transfer Characteristics, Switching Threshold, Noise Margin, Supply Voltage Scaling, Propagation Delay, Power (Dynamic, Short circuit, Leakage)

**Module-IV:** CMOS gates, Gate sizing, Capacitance estimation, Delay estimation, Logical effort, Path delay optimization, Buffer insertion, Circuit Families (Static CMOS, Ratioed gates, Cascode Voltage Switch Logic (CVSL) & Level Translators, Dynamic circuits, Pass Transistor circuits)

**Module-V:** Sequencing Elements, Sequencing Methods (Flip flop, Latch), Delay definitions, Circuit Implementations of Latch/ Flop (Static, Dynamic), Max delay constraints, Min delay constraints, Time Borrowing.

**Module-VI:** Adders (Basic terminology, Full adder circuit design, Inverting Adder, Carry Save Adder, Carry Select Adder, Carry Look Ahead Adder), Multipliers (Basic Terminology, Booth and Modified Booth Encoding, 2s Complement Arithmetic, Array Multiplier, Carry Save Multiplier, Signed multiplication and carry save implementation, Final Addition)

**Text Books:**

1. Digital Integrated Circuits: A Design Perspective" by Jan M. Rabaey.
2. Digital Integrated Circuits Jan M. Rabaey, Anantha Chandrakasan and Borivoje Nikolic 2nd Edition, Prentice Hall India.
3. CMOS VLSI Design, Neil H.E. Weste, David Harris and Ayan Banerjee, 3rd Edition, Pearson Education

**Reference Books:**

1. Principles of CMOS VLSI Design: A Systems Perspective" by Neil Weste and Kamran Eshraghian.
2. VLSI Design and EDA Tools" by Michael J. Flynn

Course Code	Course Name	L-T-P: C
EC262	IC Design Lab	0-0-2: 1

**Objectives:** The Digital IC Design Laboratory course aims to provide hands-on experience in designing and implementing digital integrated circuits. The primary objective is to familiarize students with the practical aspects of digital circuit design, from conceptualization to testing and verification. Through experiments and projects, students will gain proficiency in using industry-standard design tools such as Hardware Description Languages (HDLs), simulation software, and synthesis tools.

**Learning Outcomes:** On completion of the course, students will be able to

1. Identify the pros and cons, and the trends related to VLSI Design.
2. Create layout designs for CMOS circuits and measure the performance metrics.
3. Understand the physical aspects of CMOS, including transistor placement, routing, and the use of metal layers.
4. Use CAD tools to simulate and verify the functionality of CMOS designs.
5. Perform static timing analysis to ensure that CMOS circuits meet their timing constraints.
6. Use techniques for minimizing power consumption in CMOS circuits.
7. Understand the trade-offs between speed, power, and area.

**List of Experiments:**

1. To derive the CMOS INV gate noise margins using DC analysis.
2. To design layout of the CMOS INV and perform the RCX extraction and measure the delays.
3. To design the layout of the CMOS NAND2 and perform the RCX extraction and measure the delays.
4. To design the layout of the CMOS NOR2 and perform the RCX extraction and measure the delays.
5. To measure the dynamic and static power consumption of the CMOS NAND gate.
6. To understand the FO4 delay and measure the FO4 delay of the INV.

7. To apply the Logical effort technique to reduce the path delay.
8. To design layout of the Pass transistor AND2 and perform the RCX extraction and measure the delays.
9. To design the layout of the DOMINO NAND2 and perform the RCX extraction and measure the delays.
10. To design the layout of the Pseudo-NMOS NOR2 and perform the RCX extraction and measure the delays.

**Text/Reference Books:**

1. Neil Weste, David Harris, Ayan Banerjee, "CMOS VLSI design: A circuits and systems perspective", 3rd Edition, Pearson.
2. Sung-Mo Kang, Yusuf Leblebici, "CMOS Digital Integrated Circuits", 3rd Edition, Tata McGraw Hill.

Course Code	Course Name	L-T-P: C
EC204	Principles of Communication	3-0-3:3

**Objectives:** To study the fundamentals of communication systems. Discussions will include basic tools such as Fourier Series/ Transform, the course will also cover several important modulation techniques such as Amplitude Modulation, Frequency Modulation, Phase Modulation etc. Sampling process and Quantization, including Nyquist criterion and reconstruction of the original signal from the sampled signal. Effect of noise on signals and communication systems. Such knowledge leads to better understanding of digital and wireless communication techniques as a foundation for more advanced communication engineering subjects.

**Learning Outcomes:** On completion of the course, students will be able to

1. Understand the fundamental principles underlying various communication systems.
2. To interpret the transmitter and receiver blocks of various amplitude and phase modulation techniques.
3. To understand the principles and importance of sampling in the context of communication and characterize the effect of noise on communication systems

**Prerequisite:** Probability and Random Process, Fourier Transform, Signals.

**Contents:**

**Module-I:** Random process, correlation and Autocorrelation, Energy Spectral Density, Parseval’s Relation signals, random signals through linear systems, Gaussian random process and white noise; Shot noise and thermal noise; Noise figure and noise temperature of a two-port network, system noise calculations.

**Module-II:** Characteristics of AM and FM; Generation and detection techniques for AM, AM-DSB, SSB, NBFM, WBFM and PM; PLL and its applications in carrier acquisition and FM demodulation; Effect of noise on AM and FM systems, evaluation of SNR at detector output, Generation of Vestigial Side Band (VSB) modulated wave, Time - Domain description, Envelope detection of VSB wave plus carrier, Applications of VSB.

**Module-III:** Super heterodyne receivers and their characteristics; Different receiver architectures; RF and IF amplifiers, mixers, Basic concept of television system.

**Module-IV:** Introduction to Sampling, Spectrum of Sampled Signal, Aliasing, Nyquist Criterion, Signal Reconstruction from Sampled Signal, Pulse Amplitude Modulation, Pulse Width Modulation, Pulse Position Modulation, Quantization, Uniform Quantizers – Mid Rise and Midtread, Quantization noise, Lloyd Max Quantization Algorithm, Non uniform Quantizers, Delta Modulation, Differential Pulse Code Modulation (DPCM)

**Text Books:**

1. Haykin, S., “Communication Systems”, 4<sup>th</sup> Edition, John Wiley & Sons.
2. Proakias, J.G., and Salehi, M., “Communication Systems Engineering”, 2<sup>nd</sup> Edition, Pearson Education.
3. B.P. Lathi and Z. Ding, “Modern Digital and Analog Communication Systems,” 4th Ed., Oxford University Press, 2009.

**Reference Books:**

1. Kennedy G., Davis B., “Electronic Communication Systems”, Tata McGraw-Hill, 4<sup>th</sup> Edition.
2. Tomasi W., “Advanced Electronic Communication Systems”, Pearson/Prentice-Hall, 6<sup>th</sup> Edition.

Course Code	Course Name	L-T-P: C
EC264	Principles of Communication Lab	0-1-2: 2

**Objectives:** To offer practical experience in simulating and implementing different communication techniques—including Carrier modulation and

Demodulation, Envelope detector, mixer, FM: Modulation and demodulation, Superheterodyne receiver, Sampling, Pulse amplitude modulation, pulse width modulation, pulse position modulation—using modern software tools and laboratory equipment.

**Learning Outcomes:**

1. Analyse how random signals behave when passed through LTI systems.
2. Measure and analyze shot noise and thermal noise in circuits.
3. Generation of AM (DSB, SSB) signals and detect them using envelope detectors.
4. FM signal generation and demodulation using PLL-based detection.
5. Compare AM and FM under different noise conditions.
6. Verify the Nyquist criterion and study aliasing effects.
7. Generate and analyse PAM, PWM, and PPM waveforms.
8. Visualization of sampling, quantization, and PCM modulation.

**List of Experiments (Hardware followed by Simulations):**

1. Introduction to Communication Lab.
2. Linear Systems and Signals: To generate periodic rectangular signals of varying duty cycles, Observe the Fourier representation of the generated signals, Low-pass Filter (R-C), Convolution and System Impulse Response, NI - Analyse how random signals behave when passed through LTI systems. NI - Measure and analyze shot noise and thermal noise in circuits.
3. Fourier Transform and Properties: To verify the linearity property of Fourier Series/Transform., Superposition of two sinusoidal signals, Second order low pass Butter-worth filter to remove one of the sinusoid, Phase shifter - to shift the phase of a sinusoidal signal by  $\pi/2$ , Phase shift Vs. Time delay.
4. Generation of AM (DSB, SSB) signals and detect them using envelope detectors: Using a differential amplifier, create an analog signal multiplier to produce amplitude modulated signals. (varying  $\mu$  -[0-1], Analyze the effect of modulation index on modulation efficiency and the value of RC used in detector for a single tone modulation case, To implement a Double Sideband - Suppressed Carrier (DSB-SC) modulator using a sampler (switch) and a band-pass filter, To recover the modulating signal with the help of a coherent detector.
5. FM signal generation and demodulation using PLL-based detection.
6. Compare AM and FM under different noise conditions.

7. Verify the Nyquist criterion and study aliasing effects.
8. Generate and analyse PAM, PWM, and PPM waveforms.
9. Visualization of sampling, quantization, and PCM modulation.

**Text Books:**

1. Haykin, S., “Communication Systems”, 4th Edition, John Wiley & Sons.
2. Proakis, J.G., and Salehi, M., “Communication Systems Engineering”, 2nd Edition, Pearson Education.
3. B.P. Lathi and Z. Ding, “Modern Digital and Analog Communication Systems,” 4th Ed., Oxford University Press, 2009.

**Reference Books:**

1. Kennedy G., Davis B., “Electronic Communication Systems”, Tata McGraw-Hill, 4th Edition.
2. Tomasi W., “Advanced Electronic Communication Systems”, Pearson/Prentice-Hall, 6th Edition.

Course Code	Course Name	L-T-P: C
EC206	Control Systems	3-0-0: 3

**Objectives:** To introduce the concepts of modelling, analysis, and design of simple linear and nonlinear dynamic systems. The course covers time-domain and frequency-domain analysis, stability criteria, compensation techniques, and state-space representation to provide a strong foundation in control systems engineering.

**Learning Outcomes:**

1. Understand the fundamental principles of control systems and their classifications.
2. Develop mathematical models of dynamic systems and analyse their response.
3. Perform time-domain and frequency-domain analysis to assess system performance.
4. Apply root locus and Bode plot techniques for stability analysis.
5. Design compensators to improve system performance.
6. Utilize state-space representation for system modelling and control.
7. Analyse basic nonlinear control systems and stability criteria.

**Prerequisite:** None

**Contents:**

**Module-I:**Control System Concepts and Classification: Open loop, closed loop, continuous, discrete, linear and nonlinear control systems; Mathematical Models of Systems: Impulse response and transfer function, block-diagram model and signal flow graphs.

**Module-II:**Time Domain Analysis: Transient and steady state responses of first and second order systems, steady state errors, control of transient response; Basic control actions and their effects on transient and steady state responses; Root Locus Technique: Root loci, properties and construction of root loci, effects of adding and moving poles and zeros, root locus of conditionally stable systems, generalized root contour.

**Module-III:**Frequency Domain Analysis: Routh Hurwitz criterion, Bode and Nyquist diagrams, gain magnitude and phase shift plots, frequency domain specifications, peak resonance and resonant frequency of a second order system, gain margin and phase margin, conditionally stable system; Compensation Design in s and  $\omega$  Planes: Introduction, phase lead compensation, phase lag compensation; Design of phase-lead and phase-lag compensation by Bode plot and root locus methods.

**Module-IV:**State Variable Technique: Derivation of state model of LTI continuous time systems, state equations, state transition matrix, solution of state equations; Basic Nonlinear Analysis: Linearization, describing function and phase plane methods, stability concepts and Lyapunov functions.

**Text Books:**

1. Gopal, M., “Control Systems: Principle and Design”, 2nd Ed., Tata 2002 McGraw-Hill, 2002.
2. Kuo, B.C., “Automatic Control Systems”, 8th Ed., Wiley India, 2008.

**Reference Books:**

1. Ogata, K., “Modern Control Engineering”, 4th Ed., Pearson Education, 2008.
2. Dorf, R.C. and Bishop, R.H., “Modern Control Systems”, 11th Ed., Prentice-Hall of India, 2007.
3. Nise, N. S., “Control Systems Engineering”, 4th Ed., Wiley India, 2008.

Course Code	Course Name	L-T-P: C
MC204	Optimization Techniques	3-1-0: 4

**Objectives:** This course introduces the principal

algorithms for linear, network, discrete, nonlinear, dynamic optimization and optimal control. Emphasis is on methodology and the underlying mathematical structures. Topics include the simplex method, network flow methods, branch and bound and cutting plane methods for discrete optimization, optimality conditions for nonlinear optimization, interior point methods for convex optimization, Newton's method, heuristic methods, and dynamic programming and optimal control methods.

**Learning Outcomes:** Understand and apply Linear Optimization methods, Master discrete and Nonlinear Optimization techniques, Solve Large-Scale and Robust Optimization problems

**Prerequisites:** None

**Contents:** Introduction to Optimization, Applications of linear optimization, Geometry of linear optimization, Simplex method, Duality theory, Sensitivity analysis, Integer Programming, Branch and bound and cutting planes, Applications of discrete optimization, Lagrangian methods, Applications of nonlinear optimization, Optimality conditions and gradient methods, Line searches and Newton's method, Conjugate gradient methods, Affine scaling algorithm, Interior point methods, Dynamic programming, Large scale optimization, Network flows, Heuristics and approximation algorithms.

**Text Books:**

4. Bertsimas, Dimitris, and John Tsitsiklis. Introduction to Linear Optimization. Belmont, MA: Athena Scientific, 1997. ISBN: 9781886529199.
5. S. S. Rao. Engineering optimization- Theory and practice, New Age international Pvt. Ltd.
6. An introduction to optimization by E. K. P. Chong and S. H. Zak (Wiley)

**Reference Books:**

4. David G. Luenberger and Yinyu Ye. Linear and Nonlinear Programming. Springer, 2008, Third Edition.
5. Jorge Nosedal and Stephan Wright. Numerical Optimization. Springer, 2006. Springer Series in Operations Research.
6. Dimitri Bertsekas. Convex Optimization Theory. Athena Scientific, 2009.

## Semester-V

### Courses and Contents

Course Code	Course Name	L-T-P: C
CS301	Computer Networks	3-0-0: 3

**Objectives:** Provide essential knowledge about computer networks, as well as related concepts and theories.

**Learning Outcomes:** On successful completion of the course students will be able to:

1. Gain understanding of fundamental network theory and layered communication architectures.
2. Provide solutions to various challenges in network theory.
3. Appreciate the layered model of computer networking.
4. Identify basic protocols and design issues related to the layered model.
5. Design and implement protocols for various networking layers.

**Prerequisite:** None

#### Course Content:

**Module I:** Application Layer- This module covers the principles of network applications, including client-server and peer-to-peer models, process communication, and socket programming. It introduces key application protocols such as HTTP, FTP, SMTP, POP3/IMAP, and DNS, along with an overview of multimedia and peer-to-peer applications.

**Module II:** Transport Layer- This module focuses on transport services, multiplexing and demultiplexing, and reliable data transfer. It explains UDP and TCP, including flow control, congestion control, and TCP congestion control mechanisms.

**Module III:** Network Layer- This module introduces network layer functions such as forwarding and routing. It covers routing algorithms, IP addressing (IPv4/IPv6), subnetting, ICMP, DHCP, and an overview of Internet routing and software-defined networking.

**Module IV:** Data Link Layer- This module discusses data link layer services, error detection and correction, and multiple access protocols such as ALOHA and CSMA/CD. It also includes Ethernet, ARP, LAN switching, VLANs, and wireless LANs.

**Module V:** Physical Layer and Network Security- This module presents basic concepts of the physical layer,

transmission media, and signal transmission. It also introduces network security concepts including cryptography, authentication, SSL/TLS, firewalls, intrusion detection systems, and common attacks with countermeasures.

#### Text Books:

1. Computer Networking: A top-down approach featuring the Internet-J. F. Kurose, K. W. Ross.
2. Computer Networks, A. S. Tanenbaum.
3. Computer Networks: A Systems Approach, L. L. Peterson, B. S. Davie

#### Reference Books:

1. AS Tanenbaum, DJ Wetherall, Computer Networks, 5th Ed., Prentice-Hall, 2010.
2. LL Peterson, BS Davie, Computer Networks: A Systems Approach, 5th Ed., Morgan-Kaufman, 2011.
3. W Stallings, Cryptography and Network Security, Principles and Practice, 5th Ed., Prentice-Hall, 2010.

Course Code	Course Name	L-T-P: C
CS361	Computer Networks Lab	3-0-0: 3

#### Tentative List of Laboratory Tasks:

1. Analyze the IIITV Network structure and basic networking hardware devices such as Hub, Switch, Router, Firewall, NIC, Modem, Gateway, etc. Write down and submit the difference between various networking devices.
2. Study various types of network cables and their usage. Construct the cross and straight cables using a crimping tool. Install and configure wired and wireless NIC.
3. Configure Host IP, Subnet Mask and Default Gateway in a System in LAN (TCP/IP Configuration). Transfer files between machines in LAN using FTP. Configuration and install a Print server in a LAN and share the printer.
4. Understand and implement basic networking commands such as Ping, Ipconfig, Traceroute, Arp, Netstat, Whois, SSH, etc.
5. Implement standard error detection and correction methods.
6. Study and implement Socket Programming using any preferred internet-based programming language.
7. Study and implement flow control protocols: sliding window and go-back-n protocols.
8. Configure Internet connection and use IPCONFIG,

PING / Tracer, and Net stat utilities to debug the network issues.

9. Configure a Network topology using packet tracer software. Hands-on with Wireshark.
10. Hands-on with Cisco Packet Tracer.

**Reference:** Laboratory instructions and manuals.

Course Code	Course Name	L-T-P: C
CS303	Operating Systems	3-0-0: 3

**Objectives:** This course provides undergraduate students with knowledge about contemporary operating system design, its relationship between memory, processor, file interface, and processes, with exposure to Linux-based systems and security aspects of operating systems.

**Learning Outcomes:** On successful completion of this course, students should be able to:

1. Describe process management and concepts of threading, multitasking, IPC, and Linux process handling.
2. Differentiate scheduling algorithms and identify causes of deadlocks and their remedial measures, including security implications.
3. Describe memory management techniques, file system interfacing, and disk scheduling, with focus on protection and access control.
4. Understand OS components and analyze basic OS security mechanisms such as authentication, access control, and isolation

#### Contents:

**Module I - Introduction-** Overview and history of Operating Systems, functions of an OS, device drivers, I/O interrupts, system call interface, and introduction to Linux OS, shell basics, and security features of OS.

**Module II - Process Management-** Process abstraction, process states and PCB, threads, process scheduling, real-time scheduling, classical synchronization problems, semaphores, monitors, deadlocks and their handling, along with process isolation and security considerations in multitasking systems.

**Module III - Memory Management-** Segmentation, demand paging, hardware support, page fault handling, page replacement algorithms, shared memory, and memory protection mechanisms and secure memory management concepts.

**Module IV - Storage Management-** Disk structure, disk scheduling, disk management, RAID structure, file

organization and mounting, file descriptors, directory structure and implementation, I/O systems, along with file system security, permissions (Linux), and access control mechanisms.

#### Text Books:

1. Operating Systems: Internals and Design Principles.
2. Modern Operating Systems.

#### Reference Books:

1. Operating System Concepts.
2. Operating System Concepts Based Approach.

Course Code	Course Name	L-T-P: C
CS363	Operating Systems Lab	0-0-2: 1

#### Tentative List of Lab Exercises

1. Process creation and management using fork(), exec(), wait() in Linux.
2. Implementation of CPU scheduling algorithms (FCFS, SJF, Round Robin).
3. Synchronization problems using semaphores/mutex (Producer-Consumer, Reader-Writer).
4. Thread creation and synchronization using pthreads.
5. Deadlock detection and recovery simulation.
6. Implementation of page replacement algorithms (FIFO, LRU, Optimal).
7. Simulation of demand paging and page fault handling.
8. Linux file permissions and access control (chmod, chown, ACLs).
9. Basic security exercises: process monitoring, user management, and system protection in Linux.

Course Code	Course Name	L-T-P: C
CS305	Formal Language and Automata Theory	3-1-0: 4

**Objectives:** This course introduces students to the various types of regular languages, their equivalences to finite automata, Turing machines and Undecidability.

**Learning Outcomes:** On successful completion of this course, students should be able to:

1. Apply Languages and their principles.
2. Differentiate various computational models in theoretical computer science.

3. Analyze complexity of computational models.

**Prerequisite:** MA102.

**Contents:**

**Module I:** Mathematical Preliminaries: Review of Set theory, Functions and Relations, Graphs, Proof Techniques.

**Module II:** Finite Automata and Regular Languages: Deterministic Finite Automata (DFA), Non Deterministic Finite Automata (NFA), Equivalence between DFA and NFA, Regular Language and their relationship with Finite Automata, Regular Grammar, Properties of Regular Languages, Arden’s Theorem, Minimization of Finite Automata, Pumping Lemma for Regular Languages.

**Module III:** Push Down Automata and Context Free Languages: Push Down Automata, Context Free Languages (CFL) and Grammar (CFG), Simplification of CFG, Properties of CFG, Pumping Lemma for CFL.

**Module V:** Turing Machine and Undecidability: Introduction to Turing Machine (TM), Variants of TM, Context Sensitive Languages and Linear Bounded Automata, Recursive and Recursive Enumerable Languages, Halting Problem, Post Correspondence Problem, Undecidable Problem, Complexity Analysis.

**Text Books:**

1. *An Introduction to Formal Languages and Automata*, Peter Linz, Jones and Bartlett Publication.
2. *Introduction to Theory of Computation*, Micheal Sipser, Cengage Learning Publication.

**Reference Books:**

1. *Elements of the Theory of Computation*, H. Lewis and C. Papadimitrou, Prentice Hall Publication.
2. *Introduction to Languages and Theory of Computation*, John C. Martin, McGraw Hill Publication.
3. *Introduction to Automata Theory, Languages and Computation*, John E. Hopcraft, Rajeev Motwani, Jeffrey D. Ullman, Pearson Publication.
4. *Introduction to Formal Languages, Automata Theory and Computation*, Kamala Kirtivasan and Rama R., Pearson Publication.

Course Code	Course Name	L-T-P: C
CS307	Artificial Intelligence	3-0-0: 3

**Objectives:** The use of computers to solve complex

problems is the fundamental theme of this course. The notion of intelligence captured by problem solving ability is a starting point of the course. Understanding the difficult problems in computation and interpreting software as intelligent agents is important. Modeling the problems in a way that can be solved using computer programs is very crucial to understanding artificial intelligence. The course aims at building the foundations for advanced courses in the field of AI.

**Learning Outcomes:** Learning Outcomes: On completion of this course the student should be able to:

1. Model a real world problem in a formal way with appropriate specifications, for example: search problems, multiplayer games and strategies, expert systems etc.
2. Select suitable search strategy for the given problem.
3. Setup a deep learning computational pipeline for solving moderate size classification and regression problems.
4. Program computer to solve logical inference problems with uncertainty

**Prerequisite:** MA201, CS102, Design and Analysis of Algorithms

**Contents:**

**Module I: Introduction to AI and Search-** History, the state of the art; intelligent agents; structure; environment; Configuration and Planning Problems, State space representation, Breadth-first search; uniform cost search; depth-first search; depth-limited search; iterative, deepening search; bi-directional search; heuristic search techniques; comparing search strategies, randomized search, adversarial search, alpha-beta pruning

**Module II: Probabilistic Reasoning** - Probability and Bayes theorem; certainty factors; Bayesian networks (Graphical Models), Markov Networks; Hidden Markov Model, Decision Tree and Random Forest

**Module III: Neural Networks** - Neuron as a nonlinear combiner, Capacity of a single neuron, Feed-forward networks and Universal Approximation Theorem, Hopfield Network and Associative Memory

**Module IV: Reinforcement Learning and Computational Game Theory** - Markov Decision Process, Reinforcement Learning, Extensive form games, Social Choice and Mechanism Design

**Module V: Agentic AI Foundations-** Retrieval-augmented generation (RAG), Embeddings & vector search, Introduction to Multimodal pipelines (vision+text/audio integration patterns), Introduction to Multi-agent systems, Ethics, safety for deployed agents.

**Text Books:**

1. Artificial Intelligence – A Modern Approach, S. Russell, P. Norvig, Pearson Education, New Delhi, 2021 (Fourth Edition).
2. A First Course in Artificial Intelligence, Khemani D., Tata McGraw Hill, 2014

**Reference Books:**

1. Reinforcement Learning: An Introduction, Second Edition, Sutton and Barto MIT Press, 2018.
2. Perceptrons - An Introduction to Computational Geometry, Marvin Minsky and Seymour Papert, The MIT Press, 1969.
3. Artificial Intelligence: A guide to intelligent systems, Negnevitsky M., Pearson Education, 2005.
4. The Quest for Artificial Intelligence: A history of ideas and achievements, Nilsson N., Cambridge University Press, 2008.
5. Selected Papers on Deep Belief Networks, Multi-agent Reinforcement Learning and Markov Logic Networks.

Course Code	Course Name	L-T-P: C
CS367	Artificial Intelligence Lab	0-0-2: 1

**Objectives:** To provide hands-on experience in implementing foundational AI techniques including search, heuristic reasoning, probabilistic modeling, reinforcement learning, and sequential decision-making.

**Learning Outcomes:** Learning Outcomes: On completion of this course the student should be able to:

1. Implement and experimentally analyze classical and heuristic search algorithms for solving structured AI problems.
2. Develop and evaluate AI models for uncertainty reasoning, reinforcement learning, and sequential decision making.
3. Design and implement intelligent agents integrating search, optimization, probabilistic inference, and learning techniques.

**Prerequisite:** MA201, CS102, Design and Analysis of Algorithms

**Tentative Laboratory Assignments:**

**Lab 1:** To be able to model a given problem in terms of state space search problem and solve the same using BFS/ DFS

**Lab 2:** To design a graph search agent and understand the use of a hash table, queue in state space search.

**Lab 3:** To understand the use of Heuristic function for reducing the size of the search space. Explore non-classical search algorithms for large problems.

**Lab 4:** Non-deterministic Search | Simulated Annealing: For problems with large search spaces, randomized search becomes a meaningful option given partial/ full-information about the domain.

**Lab 5:** Understand the graphical models for inference under uncertainty, build Bayesian Network in R, Learn the structure and CPTs from Data, naive Bayes classification with dependency between features.

**Lab 6:** To understand the working of Hopfield network and use it for solving some interesting combinatorial problems

**Lab 7:** Basics of data structure needed for state-space search tasks and use of random numbers required for MDP and RL, Understanding Exploitation - Exploration in simple n-arm bandit reinforcement learning task, epsilon-greedy algorithm

**Lab 8:** Understand the process of sequential decision making (stochastic environment) and the connection with reinforcement learning.

Course Code	Course Name	L-T-P: C
CS309	Introduction to Cryptography and Network Security	3-0-0: 3

**Objectives:** The course intends to provide a theoretical knowledge and hands-on experience of cryptographic algorithms, cryptanalysis, and network security protocols, used to ensure the security of data.

**Learning Outcomes:** On successful completion of this course, students should be able to:

1. Understand the role of cryptography and cryptanalysis in security.
2. Effectively use the cryptographic algorithms to ensure the confidentiality, integrity, availability, etc. of information.
3. Effectively analyze the security strengths/ weaknesses of cryptographic algorithms.
4. Learn the network security protocols used to ensure the web application security.

**Prerequisites:** CS101, MA102, and CS301.

**Contents:**

**Module I:** Secret Key Cryptography: Substitution-Permutation network, Feistel structure, Block ciphers, Data encryption standard (DES), Advanced encryption standard (AES), Stream ciphers. Modes of Operation: Electronic code book mode

(ECB), Cipher block chaining mode (CBC), Cipher feedback mode (CFB), Output feedback mode (OFB), Counter mode (CTR).

**Module II:** Concepts of Number Theory and Finite Fields: Euclidian algorithm, Modular arithmetic, Groups, Rings, Finite fields, Polynomial arithmetic, Fermat's theorem, Euler's theorem, Chinese remainder theorem (CRT), Integer factorization problem, Discrete logarithm problem, Elliptic curve discrete logarithm problem (ECDLP).

**Module III:** Data Integrity: Cryptographic Hash functions, Secure hash algorithm (SHA2 or SHA3); Message authentication codes (MAC).

**Module IV:** Public Key Cryptography: RSA cryptosystem, Rabin's cryptosystem, Diffie-Hellman key exchange, Elgamal cryptosystem, Digital signature algorithm (DSA), Elliptic curve arithmetic, Elliptic curve digital signature algorithm (ECDSA).

**Security Models:** Ciphertext-only attacks, Known-plaintext attacks, Chosen plaintext attacks (CPA), Chosen ciphertext Attacks (CCA), Adaptive chosen ciphertext attacks (CCA2).

**Module V:** Network Security: Introduction to Web application security, Secure socket layers (SSL) / Transport layer security (TLS), HTTPS, Secure Shell (SSH), Access controls, Firewalls, Packet filtering firewall, Stateful inspection firewalls.

#### Text Books:

1. William Stallings, *Cryptography and Network Security - Principles and Practice*, Seventh Edition, Publisher: Pearson, 2016.
2. Trappe and Washington, *Introduction to Cryptography with Coding Theory*, Second Edition, Publisher: Pearson, 2009.
3. Katz & Lindell, *Introduction to Modern Cryptography: Principles and Protocols*, Second Edition, Publisher: Chapman & Hall/CRC, 2014.

#### Reference Books:

1. Bruce Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C*, Second Edition, Publisher: Wiley, 1996.
2. Douglas R. Stinson, *Cryptography: Theory and Practice*, Third Edition, Publisher: Chapman and Hall/CRC, 2005.
3. Kaufman et al., *Network Security*, Second Edition, Publisher: Prentice Hall, 2002.

Course Code	Course Name	L-T-P: C
-------------	-------------	----------

CS369	Introduction to Cryptography and Network Security Lab	0-0-2: 1
-------	---	----------

**Objectives:** The Laboratory course is intended to Implement and empirically analyze the state-of-the-art cryptographic algorithms used for encryption, authentication, digital signatures, key exchange, etc. Empirically analyze the network security protocols such as HTTPS. Effectively use the tools and techniques such as Pretty Good Privacy, etc.

**Learning Outcomes:** On successful completion of this course, students should be able to:

1. Implement and analyze the classical ciphers, symmetric-key cryptosystems, and asymmetric-key cryptosystems.
2. Encrypt and decrypt the information using symmetric-key algorithms.
3. Encrypt and decrypt the information using asymmetric-key algorithms.
4. Protect the integrity of the information using symmetric-key and asymmetric-key algorithms.
5. Perform the secure key-exchange using asymmetric-key algorithms.
6. Digitally sign and verify the information.
7. Effectively use the network security tools.

**Prerequisites:** CS101, CS161, MA102, CS263, CS301 and CS361.

#### Contents:

Laboratory assignments will consist of the following topics to develop creativity and analytical abilities of students. The assignments should be in the form of hands-on experience of cryptosystems, in one (or more) of the programming language(s), and network security protocols.

1. Classical ciphers to understand encryption, decryption, cryptanalysis.
2. Data Encryption Standard.
3. Advanced Encryption Standard, Generation of S-Boxes used in AES.
4. Modes of operations and their analysis.
5. Cryptographic hash algorithm such as SHA 512.
6. Message authentication codes such as HMAC.
7. Number theory related concepts such as Chinese remainder theorem (CRT).
8. Analyze on the impact of various parameters of size of number theoretic problems such as integer factorization problem, Discrete logarithm problem, Elliptic curve discrete logarithm problem (ECDLP).
9. RSA cryptosystem, Diffie-Hellman key exchange, Elgamal cryptosystem (1024/2048-bits).
10. Implementation of elliptic curve arithmetic.

11. Elliptic curve digital signature algorithm (ECDSA).
12. Secure socket layers (SSL)/Transport layer security (TLS).
13. Secure shell (SSH).
14. Firewalls (e.g. iptables)

**Reference:** Laboratory instructions and handouts.

Course Code	Course Name	L-T-P: C
PH301	Thermal and Statistical Physics	3-0-0: 3

**Objectives:** The course provides a comprehensive understanding of macroscopic thermodynamic laws, microscopic statistical mechanics, and their synthesis in explaining physical systems. It focuses on mastering entropy, energy potentials, heat engines, and applying statistical probability to molecular behaviors

**Learning Outcomes:** Upon successful completion of the course, students will be able to:

1. Utilize thermodynamic laws, entropy, and statistical mechanics to analyze macroscopic systems, heat engines, and particle distributions.
2. Learn to bridge microscopic quantum behavior with macroscopic properties, calculate partition functions, and apply statistical methods to solve complex problems in physics.

**Prerequisites:** PH202

**Contents:**

**Module-I:** Temperature and Zeroth Law of Thermodynamics, Work, Heat and First Law of Thermodynamics, Ideal Gas and Heat Capacities, Second Law of Thermodynamics, Carnot Cycle, Entropy, Thermodynamic variables and energies.

**Module-II:** Review of Probability and Distributions, One-dimensional Random Walk Problem, Binomial and Gaussian Distributions, Large N limit, Brownian Motion and Origin of Dissipation, Percolation problem, Markov processes

**Module-III:** Fundamentals of Statistical Mechanics: Introduction to Microcanonical Ensemble, Canonical Ensemble and Grand Canonical Ensemble, Partition Function, Kinetic Theory of Gas, Maxwell-Boltzmann distribution

**Module-IV:** Quantum Statistical Systems: von-Neumann Entropy, Bosonic and Fermionic systems, Bose-Einstein and Fermi-Dirac statistics, Ideal Bose and Fermi gas, Pauli Paramagnetism, Landau Diamagnetism

**Module-V:** Phase transitions: First Order and Second Order Phase transitions, Liquid-Gas transition, Landau theory of Phase Transitions, Superconductivity, Superfluidity

**Text Books:**

1. F Reif, "Fundamentals of Statistical and Thermal Physics".
2. R K Pathria, "Statistical Mechanics"

**Reference Books:**

1. Kerson Huang, "Statistical Mechanics"

Course Code	Course Name	L-T-P: C
PH303	Computational Physics	3-0-0: 3

**Objectives:** This course aims to equip students with a solid foundation in computational techniques for modeling and analyzing materials at the atomic and molecular levels. It focuses on developing proficiency in Monte Carlo and Molecular Dynamics simulations, enabling students to apply numerical methods to investigate structural, thermodynamic, and dynamic properties of materials.

**Learning Outcomes:** By the end of this course, students will be able to:

1. Understand and apply the fundamental principles of classical, statistical, and quantum mechanics in computational materials modeling.
2. Implement and analyze Monte Carlo and Molecular Dynamics simulations for studying material properties and phase behavior.
3. Use random number generation and numerical integration methods in deterministic and stochastic simulations.
4. Interpret simulation data to evaluate structural and thermodynamic properties of materials.

**Prerequisites:** Classical Mechanics, Quantum Mechanics, Statistical Physics, and Solid State Physics

**Contents:**

**Module-I:** Introduction: Computer simulations and problems in material science, Brief review of classical mechanics, statistical mechanics and quantum mechanics as a starting point. Random Number generation and testing, Generation of random numbers with Uniform, Gaussian, and exponential distribution, Numerical Integration: (a) Deterministic: Trapezoidal and Simpson's method & (b) Multi-dimensional Integration using stochastic methods.

**Module-II:** Monte Carlo simulations: Importance sampling and the metropolis method, basic Monte Carlo algorithm, trial moves, random number generators, estimators. Application: solid-liquid phase-transition in the Lennard-Jones fluid and the magnetic transition in the Ising model. Advanced applications–Monte Carlo in various ensembles, Kinetic Monte Carlo, Monte Carlo methods for rigid molecules and polymers.

**Module-III:** Molecular Dynamics: Basic principles of molecular dynamics (MD), numerical integration of equations of motion: Verlet and Velocity-Verlet algorithms, Classical force-fields: bonded and non-bonded interactions, parameterization of force-fields. Applications: Calculation of diffusion constants and radial distribution functions for a Lennard-Jones fluid using the Andersen thermostat. Advanced applications: MD in various ensembles (NVE, NPT, NVT), thermostats and baro-stat techniques, and constrained MD.

**Module-IV:** Computational Techniques and Best Practices: Neighbor lists, Multiple time-step algorithms, Methods for handling long-range interactions, including Ewald summation and cutoff schemes.

**Module-V:** Advanced Simulation Techniques: Biased Monte Carlo Schemes, Rare Event, Brownian dynamics, Dissipative particle dynamics.

**Text Books:**

1. D. Frenkel and B. Smit, Understanding Molecular Simulations (ed. 2).
2. A. R. Leach, Molecular Modeling: Principles and Applications.
3. J. M. Thijssen, Computational Physics.
4. T. Pang, An Introduction to Computational Physics

**Reference Books:**

1. Nicholas J. Giordano & Hisao Nakanishi, Computational Physics.
2. R. C. Verma, P. K. Ahluwalia, K. C. Sharma, Computational Physics: An Introduction.
3. M. E. J. Newman & G. T. Barkema, Monte Carlo Methods in Statistical Physics.

Course Code	Course Name	L-T-P: C
PH303	Computational Physics Lab	0-0-2: 1

**Objectives:**

1. To introduce students to the fundamental computational methods such as random number generation, numerical integration, Monte Carlo

simulations, and Molecular dynamics for solving real-life physics problems.

2. To develop the ability to model and simulate physical systems, including Ising model, Phase transitions, Lennard–Jones fluids, and One-dimensional chain, using both self-written programs and advanced simulation tools like LAMMPS.
3. To train students in analyzing thermodynamic, structural, and dynamical properties of simulated systems through statistical techniques and data analysis methods.
4. To build computational problem-solving skills required for modern research in statistical physics and materials science.

**Learning Outcomes:** From this laboratory, students will learn to implement numerical algorithms, generate random numbers, and apply Monte Carlo and Molecular Dynamics methods to physical systems. They will gain hands-on experience with simulating models such as the Ising model and Lennard–Jones fluids, and analyze their structural, thermodynamic, and dynamical properties using both custom code and open-source package LAMMPS. Overall, the lab builds a strong foundation in computational techniques widely used in modern physics and materials science.

**Prerequisites:** MATLAB/Python

**Contents:**

1. Random Number Generation and Statistical Analysis Using a Linear Congruential Generator.
2. Comparative Study of Deterministic and Monte Carlo (MC) Numerical Integration.
3. Monte Carlo Simulation and Analysis of the 2D Ising Model.
4. Temperature-Driven Phase Transitions in the Ising Model (Monte Carlo Analysis).
5. Monte Carlo Simulation of a 3D Lennard–Jones Fluid.
6. Investigation of the Solid–Liquid Phase Transition in a Lennard–Jones System.
7. Introduction to LAMMPS: Lennard–Jones Fluid Simulation.
8. Molecular Dynamics in NVE, NVT, and NPT Ensembles Using LAMMPS.
9. Radial Distribution Function and Diffusion Coefficient Analysis Using LAMMPS.
10. Vibrational Modes of a Linear Chain using LAMMPS

**Text Books:**

1. M. P. Allen & D. J. Tildesley, Computer Simulation of Liquids.
2. LAMMPS Manual.

3. David P. Landau & Kurt Binder, A Guide to Monte Carlo Simulations in Statistical Physics.

Course Code	Course Name	L-T-P: C
AI301	Control Systems	3-1-0:4

**Objectives:** The course aims to enable students to understand the fundamental concepts of control systems and feedback mechanisms. The students will be able to model physical systems mathematically, analyze time-domain and frequency-domain behaviour, determine system stability, and design basic controllers to meet specified performance requirements in engineering applications.

**Learning Outcomes:** On completion of this course, the student should be able to:

1. Understand and explain the basic concepts, terminology, and applications of control systems.
2. Develop mathematical models and block diagrams for electrical and mechanical systems.
3. Analyze the time response, frequency response, and stability of linear control systems.
4. Apply classical control techniques for controller design and performance improvement.
5. Introduce state-space concepts and analyse system controllability and observability.

**Prerequisite:** Linear Systems and Signal Processing

**Contents:**

#### Module-1: Introduction and Mathematical Modeling

Control system concepts; open-loop and closed-loop systems; feedback and its advantages; examples of control systems; mathematical modeling of electrical and mechanical systems; differential equations; transfer function; block diagram representation; signal flow graph; Mason's gain formula.

#### Module-2: Time Domain Analysis

Standard test signals; time response of first-order and second-order systems; transient and steady-state response; performance specifications; steady-state error; static error constants; effect of system parameters on time response.

#### Module-3: Stability Analysis and Root Locus

Concept of stability; types of stability; Routh-Hurwitz stability criterion; relative stability; root locus technique; construction of root locus; effect of poles and zeros; design using root locus method.

#### Module-4: Frequency Domain Analysis

Frequency response of control systems; polar plots; Bode plots; gain margin and phase margin; Nyquist stability criterion; closed-loop frequency response; relative stability using frequency-domain techniques.

#### Module-5: Control System Design and State Space Analysis

Controllers: proportional (P), proportional-integral (PI), proportional-derivative (PD), and PID controllers; controller design using frequency response and root locus methods; introduction to state-space modeling; state equations; state transition matrix; controllability and observability; pole placement technique (introductory).

**Text Books:**

1. I. J. Nagrath and M. Gopal, *Control Systems Engineering*, New Age International Publishers, New Delhi.
2. B. C. Kuo and F. Golnaraghi, *Automatic Control Systems*, Wiley India.

**Reference Books:**

1. K. Ogata, *Modern Control Engineering*, Pearson Education.
2. Norman S. Nise, *Control Systems Engineering*, Wiley.
3. Richard C. Dorf and Robert H. Bishop, *Modern Control Systems*, Pearson.

Course Code	Course Name	L-T-P:C
AI303	Deep Learning	3-0-2:4

**Description:** This course provides a comprehensive journey through the evolution of neural networks, starting from biological inspirations to the state-of-the-art architectures used in modern AI. It balances rigorous mathematical foundations with practical programming applications in vision, language, and generative modeling.

**Objectives:** To enable students to understand the mechanics of deep learning models, master optimization algorithms, and develop the ability to architect specialized neural networks for complex real-world data like images and sequences.

**Learning Outcomes:** Students will be able to derive and implement backpropagation, design deep architectures including CNNs and LSTMs, apply advanced regularization and optimization techniques, and explore generative models such as GANs and Variational Autoencoders.

**Prerequisite:** MA201, Machine Learning

**Contents:**

**Module I: Foundations and Feedforward Networks**

History of Deep Learning and its success stories. McCulloch Pitts Neuron, Thresholding Logic, and Perceptrons. Perceptron Learning Algorithm and Convergence. Multilayer Perceptrons (MLPs) and their representation power. Sigmoid Neurons, Feedforward Neural Networks, and the Backpropagation algorithm.

**Module II: Optimization and Regularization**

Gradient Descent (GD) and its variants: Momentum-based GD, Nesterov Accelerated GD, Stochastic GD, AdaGrad, RMSProp, and Adam. Regularization strategies: Bias-Variance tradeoff, L2 regularization, Early stopping, Dataset augmentation, Parameter sharing, and Dropout. Greedy Layerwise Pre-training and Batch Normalization.

**Module III: Feature Extraction and Computer Vision**

Linear Algebra refresher: Eigenvalues, Eigenvectors, PCA, and SVD. Autoencoders: Denoising, Sparse, and Contractive variants. Convolutional Neural Networks (CNNs): LeNet, AlexNet, ZF-Net, VGGNet, GoogLeNet, and ResNet. Applications: Object Detection (RCNN, Fast RCNN, Faster RCNN, YOLO) and Visualizing CNNs (Guided Backpropagation, Deep Dream).

**Module IV: Sequence Modeling and Attention**

Recurrent Neural Networks (RNNs) and Backpropagation Through Time (BPTT). Vanishing and Exploding Gradients. Gated Recurrent Units (GRUs) and Long Short Term Memory (LSTM) cells. Learning Vectorial Representations of Words (Word2Vec). Encoder-Decoder Models and Attention Mechanisms (Attention over images, Hierarchical Attention).

**Module V: Graphical and Generative Models**

Directed Graphical Models and Markov Networks. Restricted Boltzmann Machines (RBMs), Unsupervised Learning, and Gibbs Sampling. Variational Autoencoders (VAEs). Autoregressive Models: NADE, MADE, PixelRNN. Generative Adversarial Networks (GANs).

**Text Books:**

1. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
2. Nielsen, M. A. (2015). *Neural Networks and Deep Learning*. Determination Press.

**Reference Books:**

1. Schmidhuber, J. (2015). Deep Learning in Neural Networks: An Overview. *Neural Networks Journal*.

2. Mikolov, T., et al. (2013). Efficient Estimation of Word Representations in Vector Space. (Word2Vec Paper).

Course Code	Course Name	L-T-P: C
AI305	Natural Language Processing	3-0-2:4

**Description:** This course provides a comprehensive introduction to Natural Language Processing, covering both linguistic foundations and modern machine learning approaches. It explores techniques for analyzing, understanding, and generating human language using computational methods. The course also emphasizes recent advances such as deep learning and transformer-based models for real-world NLP applications.

**Objectives:** The course aims to develop a strong understanding of core NLP tasks such as tagging, parsing, and semantic analysis. It introduces students to statistical and machine learning techniques used in language modeling and text processing. Additionally, the course prepares learners to design and implement NLP systems using modern deep learning frameworks.

**Learning Outcomes:** After completing the course, students will be able to:

1. Understand linguistic structures and their computational representations.
2. Implement NLP pipelines including tokenization, tagging, and parsing.
3. Apply machine learning algorithms for text classification and sequence labeling.
4. Utilize word embeddings and deep learning models (RNNs, Transformers).
5. Develop NLP applications such as machine translation, question answering, and chatbots

**Prerequisite:** MA201, CS102, Machine Learning

**Contents:**

**Module I:** Foundations of NLP and Text Processing - Introduction to NLP, linguistic levels (morphology, syntax, semantics), text preprocessing, tokenization, normalization, n-grams, language modeling

**Module II:** Sequence Labeling and Parsing - Part-of-speech tagging, named entity recognition, Hidden Markov Models (HMM), Conditional Random Fields (CRF), syntactic parsing (dependency and constituency parsing)

**Module III:** Machine Learning and Representation in NLP - Text classification, Naive Bayes, logistic

regression, feature engineering, word embeddings (Word2Vec, GloVe), neural networks, recurrent neural networks (RNN, LSTM, GRU)

**Module IV:** Advanced NLP and Applications - Attention mechanisms, transformer models (BERT, GPT), transfer learning, machine translation, question answering systems, dialogue systems, information extraction.

**Tentative Laboratory Assignments:**

**Lab 1:** Text Preprocessing and Tokenization

Implement text cleaning, tokenization, stopword removal, stemming, and lemmatization using Python libraries (NLTK / spaCy). Generate n-grams and visualize word frequency distributions.

**Lab 2:** Part-of-Speech Tagging and Named Entity Recognition. Perform POS tagging using rule-based and statistical approaches. Implement Named Entity Recognition (NER) using pre-trained models (spaCy / Hugging Face). Evaluate tagging accuracy using standard datasets.

**Lab 3:** Text Classification using Machine Learning

Build a text classification model (e.g., spam detection or sentiment analysis). Use algorithms like Naive Bayes, Logistic Regression, or SVM. Perform feature extraction using Bag-of-Words and TF-IDF

**Lab 4:** Word Embeddings and Neural Models. Train or use pre-trained embeddings (Word2Vec / GloVe). Implement a simple neural network or LSTM for sequence prediction. Visualize embeddings using dimensionality reduction (t-SNE / PCA).

**Lab 5:** Transformer-based NLP Application. Use pre-trained transformer models (e.g., BERT / GPT via Hugging Face). Perform tasks such as text classification, question answering, or summarization. Fine-tune a model on a small dataset and evaluate performance

**Text Books:**

1. Jurafsky, D., & Martin, J. H. (2023). *Speech and Language Processing* (3rd ed., draft). Pearson.
2. Eisenstein, J. (2019). *Introduction to Natural Language Processing*. MIT Press.

**Reference Books:**

1. Goldberg, Y. (2017). *Neural Network Methods for Natural Language Processing*. Morgan & Claypool.
2. Vaswani, A., et al. (2017). *Attention Is All You Need*. *Advances in Neural Information Processing Systems* (NeurIPS).

Course Code	Course Name	L-T-P: C
AI307	Computer Vision	3-0-2:4

**Description:** This course introduces the fundamental concepts and techniques used in Computer Vision for analyzing and understanding visual data. It covers image formation, feature extraction, and machine learning methods for visual recognition. The course also explores modern deep learning approaches for solving real-world vision problems.

**Objectives:** The course aims to build a strong foundation in image processing and visual feature representation. It introduces classical and modern algorithms for object detection, recognition, and scene understanding. Additionally, it prepares students to design and implement computer vision systems using deep learning techniques.

**Learning Outcomes:** After completing the course, students will be able to:

1. Understand image formation and camera models.
2. Apply feature detection and image processing techniques.
3. Implement object detection and image segmentation algorithms.
4. Utilize deep learning models (CNNs) for vision tasks.
5. Develop real-world applications such as face recognition and scene understanding.

**Prerequisites:** MA201

**Contents:**

**Module I:** Image Formation and Fundamentals - Image formation, pinhole camera model, perspective projection, image filtering, edge detection, image transformations

**Module II:** Feature Detection and Image Representation - Corner detection (Harris), SIFT, SURF, feature matching, histogram-based methods, texture representation

**Module III:** Object Detection and Segmentation - Image segmentation, clustering methods (k-means), region-based methods, object detection, motion analysis

**Module IV:** Deep Learning for Computer Vision - Convolutional Neural Networks (CNNs), transfer learning, object detection models (R-CNN, YOLO), image classification, semantic segmentation

**Tentative Laboratory Assignments:**

**Lab 1:** Image Processing and Filtering

Implement image transformations (grayscale, resizing, smoothing). Apply edge detection (Sobel, Canny) using OpenCV.

**Lab 2:** Feature Detection and Matching

Detect keypoints using Harris/SIFT. Perform feature matching between images. Visualize correspondences.

**Lab 3:** Image Segmentation. Implement k-means clustering for segmentation. Apply thresholding and region-growing techniques. Evaluate segmentation quality.

**Lab 4:** Object Detection using Classical Methods

Implement sliding window detection. Use HOG features with SVM classifier. Evaluate detection accuracy

**Lab 5:** Deep Learning for Vision. Train a CNN for image classification (e.g., CIFAR-10). Use transfer learning (ResNet / VGG). Implement object detection using YOLO or Faster R-CNN

**Text Books:**

1. Szeliski, R. (2022). *Computer Vision: Algorithms and Applications* (2nd ed.). Springer.
2. Forsyth, D. A., & Ponce, J. (2012). *Computer Vision: A Modern Approach* (2nd ed.). Pearson.

**Reference Books:**

1. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
2. Prince, S. J. D. (2012). *Computer Vision: Models, Learning, and Inference*. Cambridge University Press.

Course Code	Course Name	L-T-P: C
CC301	Web Technologies	3-0-2: 4

**Objectives:** This course introduces fundamental concepts of web development, covering HTML, CSS, and JavaScript for creating interactive web pages. It explores front-end and back-end technologies, including basic server-side scripting and databases. The course emphasizes hands-on practice in building responsive and dynamic websites.

**Learning Outcomes:** On successful completion of this course, students should be able to:

1. Gain knowledge of fundamental web technologies, including HTML, CSS, and JavaScript, for creating structured and styled web pages.

2. Learn to enhance user experience using JavaScript for dynamic and interactive web content.
3. Apply CSS frameworks and techniques to build mobile-friendly and visually appealing web pages.
4. Understand the basics of server-side scripting and database connectivity for dynamic web applications.

**Prerequisite:** CS101 and CS301

**Contents:**

**Module 1** - Introduction to Web Programming: Overview of the web and how it works, client-side vs. server-side programming, introduction to web browsers and developer tools

**Module 2** - HTML – Structuring Web Pages: HTML syntax and document structure, HTML elements: headings, paragraphs, lists, tables, and forms, hyperlinks, images, and multimedia integration, semantic HTML and best practices

**Module 3** - CSS – Styling Web Pages: Introduction to CSS and its role in web design, CSS selectors, properties, and units, box model, flexbox, and grid layout, responsive design with media queries

**Module 4** – JavaScript: Basics of JavaScript: variables, data types, and operators, functions, loops, and conditionals, DOM manipulation and event handling

**Module 5** - Web Forms and User Input: Creating and styling forms, form validation using HTML5 and JavaScript, handling user input and feedback

**Module 6** - Introduction to Frontend Frameworks: Basics of Bootstrap or Tailwind CSS for faster development, introduction to JavaScript libraries (e.g., React)

**Module 7** - Basics of Backend Development: Understanding how the web server works, introduction to server-side programming (e.g., Node.js), basics of databases and SQL, sending and receiving data using HTTP requests

**Module 8** - Web Hosting and Deployment: Introduction to web hosting services, deploying websites using GitHub Pages, Netlify, or FTP, basics of domain names and web hosting.

**Text Books:**

1. HTML and CSS: Design and Build Websites, Jon Duckett, Wiley (2011).
2. Learning Web Design, Jennifer Robbins, O'Reilly (2018).

**Reference Books:**

1. JavaScript: The Definitive Guide, David Flanagan, O'Reilly (2020).

2. Web Development and Design Foundations with HTML5, Terry Felke-Morris, Pearson (2018).

### Laboratory Assignments

#### Project/Assignment-1 (Information Flow):

Implementation of complete website; PHP backend; MySQL Database; front-end Form development (text, email, radio, checkbox, select/data list)

#### Project/Assignment-2 (Validation and Structure):

Client-side validation of project/ assignment 1; Server- side validation of project/ assignment 1; Object-oriented designing of PHP backend (following MVC architecture); Unit testing; Using Git; Using GitHub.

#### Project/Assignment-3 (Session Management):

Session Management addition to project/assignment 2; User login addition to project/assignment 2; Styling & Layout addition project/assignment 2.

Course Code	Course Name	L-T-P: C
CC303	Operating Systems and Linux for Cybersecurity	3-0-2: 4

**Objectives:** This course provides undergraduate students with knowledge about contemporary operating system design, its relationship between memory, processor, file interface, and processes, with exposure to Linux-based systems and security aspects of operating systems.

**Learning Outcomes:** On successful completion of this course, students should be able to:

1. Describe process management and concepts of threading, multitasking, IPC, and Linux process handling.
2. Differentiate scheduling algorithms and identify causes of deadlocks and their remedial measures, including security implications.
3. Describe memory management techniques, file system interfacing, and disk scheduling, with focus on protection and access control.
4. Understand OS components and analyze basic OS security mechanisms such as authentication, access control, and isolation

### Contents:

**Module I** - Introduction- Overview and history of Operating Systems, functions of an OS, device drivers, I/O interrupts, system call interface, and introduction to Linux OS, shell basics, and security features of OS.

**Module II** - Process Management- Process abstraction, process states and PCB, threads, process scheduling,

real-time scheduling, classical synchronization problems, semaphores, monitors, deadlocks and their handling, along with process isolation and security considerations in multitasking systems.

**Module III** - Memory Management- Segmentation, demand paging, hardware support, page fault handling, page replacement algorithms, shared memory, and memory protection mechanisms and secure memory management concepts.

**Module IV** - Storage Management- Disk structure, disk scheduling, disk management, RAID structure, file organization and mounting, file descriptors, directory structure and implementation, I/O systems, along with file system security, permissions (Linux), and access control mechanisms.

### Text Books:

1. Operating Systems: Internals and Design Principles.
2. Modern Operating Systems

### Reference Books:

1. Operating System Concepts.
2. Operating System Concepts Based Approach

### Tentative List of Lab Exercises

1. Process creation and management using fork(), exec(), wait() in Linux.
2. Implementation of CPU scheduling algorithms (FCFS, SJF, Round Robin).
3. Synchronization problems using semaphores/ mutex (Producer-Consumer, Reader-Writer).
4. Thread creation and synchronization using pthreads.
5. Deadlock detection and recovery simulation.
6. Implementation of page replacement algorithms (FIFO, LRU, Optimal).
7. Simulation of demand paging and page fault handling.
8. Linux file permissions and access control (chmod, chown, ACLs).
9. Basic security exercises: process monitoring, user management, and system protection in Linux.

Course Code	Course Name	L-T-P: C
CC305	Cybersecurity Threats and Defense Mechanisms	3-0-2: 4

**Objectives:** To provide students with a comprehensive understanding of modern cybersecurity threats and corresponding defense mechanisms, with emphasis on web security, secure communication, and practical attack mitigation techniques.

**Learning Outcomes:** After completing this course, students will be able to:

1. Understand fundamental cybersecurity concepts and threat landscapes.
2. Identify and analyze common web-based attacks (XSS, SQL Injection, CSRF, etc.).
3. Evaluate vulnerabilities in web applications and systems.
4. Apply appropriate defense mechanisms and secure coding practices.
5. Understand secure communication protocols like SSL/TLS.
6. Design basic secure systems and mitigation strategies

**Contents:**

**Module I:** Introduction to Cybersecurity: Basics of cybersecurity: CIA triad (Confidentiality, Integrity, Availability), Threats, vulnerabilities, and risk, Types of attackers (insiders, outsiders, APTs), Attack surface and threat modeling, Malware types: virus, worm, trojan, ransomware, Cybersecurity frameworks and standards overview.

**Module II:** Web Security Fundamentals: Architecture of web applications (client-server model), HTTP/HTTPS protocol basics, Session management and authentication, Cookies: types, attributes, security issues, Same-Origin Policy and browser security model.

**Module III:** Common Web Attacks: Cross-Site Scripting (XSS): stored, reflected, DOM-based, SQL Injection (SQLi): types and exploitation, Cross-Site Request Forgery (CSRF), Command injection and file inclusion vulnerabilities, Session hijacking and fixation.

**Module IV:** Security Protocols and Mechanisms: SSL/TLS protocols and secure communication, Digital certificates and Public Key Infrastructure (PKI), Encryption basics: symmetric and asymmetric, Hashing and message authentication, Secure authentication mechanisms (2FA, OTPs).

**Module V:** Defense Mechanisms and Best Practices: Input validation and output encoding, Secure coding practices, Web Application Firewalls (WAF), Intrusion Detection and Prevention Systems (IDS/IPS), Vulnerability assessment and penetration testing basics, Security testing tools.

**Textbooks:**

1. Computer Security: Principles and Practice, William Stallings and Lawrie Brown, Computer Security: Principles and Practice, 4th ed., Pearson, 2018.
2. Web Application Hacker's Handbook, Dafydd Stuttard and Marcus Pinto, The Web Application Hacker's Handbook, Wiley.

3. Hacking: The Art of Exploitation, Jon Erickson, Hacking: The Art of Exploitation, No Starch Press.

**Reference Books:**

1. Applied Cryptography, Bruce Schneier, Wiley.
2. Security Engineering, Ross Anderson, Wiley.

Course Code	Course Name	L-T-P: C
CC307	Fundamentals of Machine Learning	3-0-2: 4

**Objectives:** Understand and utilize the concepts of machine learning for data science and electrical engineering. Focus on tools for multivariate data analysis and how to handle uncertain data with probability models.

**Learning Outcomes:** On successful completion of the course students will be able to demonstrate:

1. Generate, analyze and interpret data summaries.
2. Carry out analysis on machine learning algorithms.
3. Design and implement classifiers for machine learning applications.
4. Design and implement frequent item set mining systems.

**Prerequisite:** None

**Contents:**

**Module-I: Introduction to Machine learning:** Introduction to types of learning in machine learning and general terminology, Features of Machine Learning, Need for Machine Learning, Goals of Machine Learning, Applications, Challenges of Machine Learning, Data and Its Processing, Types of data in datasets, Data splitting in Machine Learning, Different data Pre-processing techniques for machine learning, Overview of Data Cleaning, Handling Imbalanced Data in Machine Learning, Feature Scaling, Overfitting, Underfitting, Machine learning Life Cycle

**Module-II: Supervised learning:** Supervised Learning Works, Steps Involved in Supervised Learning, Types of supervised Machine Learning, Classification and its types, Logistic Regression in Machine Learning, K-Nearest Neighbour (KNN) Algorithm, Support Vector Machine Algorithm, Naive Bayes Classifier Algorithm, Regression and its types, Linear Regression, Simple Linear Regression, Multiple Linear Regression, Polynomial Regression

**Module-III: Unsupervised learning:** Unsupervised Learning Works, Clustering in Machine Learning, Types of Clustering Methods, K

means Clustering, Mean-shift Algorithm, Mean-shift Algorithm, DBSCAN Algorithm, Association Rule Learning, Types of Association Rule Learning, Apriori Algorithm, F-P Growth Algorithm, Reinforcement learning

**Module-IV: Model selection:** Model selection criteria, Description length, feature selection, Combining classifiers, boosting, Bagging, margin, and complexity, Margin and generalization (EM) algorithm, EM, regularization, clustering, Clustering, Spectral clustering, Markov models, Hidden Markov models (HMMs), Bayesian networks, Learning Bayesian networks, Probabilistic inference, Current problems in machine learning.

**Module-V: Model Evaluation:** Performance Metrics: Accuracy, precision, recall, F1-score, ROC curve, AUC. Evaluation Techniques: Cross-validation (K-fold), train/test splits, bias-variance tradeoff. Hyperparameter Tuning: Grid search, random search, and regularization techniques.

**Text Books:**

1. "Learning From Data", Yaser S. Abu-Mostafa, Malik Magdon-Ismail, and Hsuan-Tien Lin
2. "Machine Learning", Tom Mitchell

**Reference Books:**

1. Hastie, Tibshirani, and Friedman: The Elements of Statistical Learning.
2. Christopher Bishop: Pattern Recognition and Machine Learning.
3. Kevin Murphy: Machine Learning: a Probabilistic Perspective
4. Shai Shalev-Shwartz and Shai Ben-David: Understanding Machine Learning: From Theory to Algorithms.
5. David Barber: Bayesian Reasoning and Machine Learning.
6. Richard S. Sutton and Andrew G. Barto: Reinforcement Learning: An Introduction.

Course Code	Course Name	L-T-P: C
CC309	Embedded Systems and Internet of Things	3-0-2: 4

**Objectives:**

1. To introduce the fundamentals of embedded systems and IoT architectures.
2. To understand hardware–software interaction in resource-constrained devices.

3. To analyze security threats in embedded and IoT environments.
4. To design secure IoT-based applications using appropriate protocols and protection mechanisms.

**Learning Outcomes:**

After completing the course, students will be able to:

1. Explain the architecture and components of embedded and IoT systems.
2. Develop basic embedded/IoT applications using sensors, controllers, and communication modules.
3. Identify vulnerabilities and cyber threats in IoT and embedded environments.
4. Apply cryptographic and security mechanisms to design secure IoT systems.
5. Evaluate real-world IoT cybersecurity risks and mitigation strategies.

**Contents:**

Introduction to Embedded Systems. Definition and characteristics of embedded systems. Microcontrollers vs microprocessors. Embedded system architecture and components. Memory, I/O interfaces, interrupts. Real-time operating systems (RTOS) basics. Applications in smart devices and critical infrastructure

Embedded System Design and Programming. Embedded software development lifecycle. Programming concepts for embedded systems. Interfacing sensors and actuators. Communication interfaces: UART, SPI, I2C, CAN. Power management and resource constraints. Case studies of secure embedded applications.

Fundamentals of the Internet of Things. IoT concept, evolution, and architecture. IoT layers and components. Smart sensors, gateways, and cloud integration. Communication technologies: WiFi, Bluetooth LE, Zigbee, LPWAN. IoT protocols: MQTT, CoAP, HTTP, REST APIs. IoT application domains (smart home, healthcare, industry, transport)

Security Issues in Embedded and IoT Systems. IoT threat landscape and attack surfaces. Hardware attacks: side-channel, tampering, firmware extraction. Network attacks: spoofing, MITM, replay, DoS/DDoS. Malware, botnets, and insecure firmware updates. Privacy risks and data leakage in IoT. Security requirements: confidentiality, integrity, availability, authentication.

Secure IoT and Embedded System Design. Lightweight cryptography for IoT. Authentication and key management in constrained devices. Secure boot and trusted firmware updates. Secure communication protocols (TLS/DTLS basics). Access control and identity management for IoT. Blockchain and PUF-based security concepts (introductory level).

Emerging Trends and Case Studies. Industrial IoT (IIoT) security challenges. Edge computing and fog security. AI-enabled IoT security monitoring. National/international IoT security guidelines and standards. Case studies of major IoT security breaches

**Text Books:**

1. Marwedel, Peter. Embedded System Design: Embedded Systems Foundations of Cyber-Physical Systems, and the Internet of Things. 4th ed., Springer, 2021.
2. Bahga, Arshdeep, and Vijay Madiseti. Internet of Things: A Hands-On Approach. VPT/CreateSpace Inc., 2014.
3. Relevant Research Papers

**Practical / Laboratory Experiments**

1. **To study the embedded system development workflow** using Keil IDE and simulate basic microcontroller circuits in Proteus.
2. **To design and implement an LED control program** for understanding GPIO operations, and verify the circuit using Proteus simulation and Arduino hardware.
3. **To interface a 16×2 LCD display** with a microcontroller/Arduino and develop a program to display system messages and security alerts.
4. **To design a keypad-based authentication system** using a 4×4 keypad and LCD to allow access only to authorized users.
5. **To implement a secure password verification mechanism** that locks the system and generates alerts after multiple incorrect attempts.
6. **To interface the DHT11 temperature–humidity sensor** and display real-time environmental data while validating sensor readings for reliability.
7. **To develop a sound-based intrusion detection system** that senses abnormal sound levels and triggers visual alerts.
8. **To design a flame detection and alert system** that identifies fire events and provides immediate notification using LEDs/LCD.
9. **To simulate embedded sensor-based systems in Proteus** before deploying them on hardware for secure design validation.
10. **To implement serial communication between embedded devices** and ensure correct and verified data transmission.
11. **To design a multi-sensor secure monitoring system** integrating temperature, sound, and flame sensors with centralized alert display.
12. **To develop a cybersecurity-focused embedded mini project** (such as secure smart lock, intrusion detector, or environmental monitoring node) including circuit simulation in Proteus, programming in Keil/Arduino IDE, and hardware implementation.

13. Mini-project: Design a **secure IoT prototype** (smart lock, secure sensor node, intrusion alert, etc.)

Course Code	Course Name	L-T-P: C
EC301	Computer Networks	3-0-0: 3

**Objectives:** This course develops an understanding of modern network architectures from a design and performance perspective. It clarifies network terminology and provides an opportunity to do network programming using TCP/IP. The course exposes students to emerging technologies and their potential impact.

**Learning Outcomes:**

1. Understand the TCP/IP layered model for data communication.
2. Analyse the requirements for a given organizational network layout and give the most appropriate networking architecture and technologies suited.
3. Have a working knowledge of connectionless and connection-oriented protocols.

**Prerequisite:** Introduction to Programming and Data Structures, Analog communication, Digital communication.

**Contents:**

**Module-I:** Overview of OSI reference model, Overview of an internet, internet as a service, internet architecture, circuit switching, packet switching, network performance metrics (delay, packet loss, and throughput), layered approach (TCP/IP and OSI models).

**Module-II:** Link Layer: Multiple access protocols (channel partitioning protocols, random access protocols, and CSMA protocols), Ethernet – IEEE 802.3, Token ring – IEEE 802.5, WiFi – IEEE 802.11, reliable link layer protocols (stop and wait, sliding window protocols), switches and bridges.

**Module-III:** Network Layer: IP addressing: IPv4, IPv6, and ICMP header formats; intra-domain routing: distance vector and link state routing protocols; inter-domain routing: BGP; routing for multicasting and broadcasting.

**Module-IV:** Transport Layer: Principles of reliable data transfer; connection-oriented transport: TCP connection establishment, TCP timeout estimation, TCP RTT

estimation, TCP congestion control; connectionless transport: UDP.

**Module-V:** Application layer: network applications, hypertext transfer protocol, domain name system, simple mail transfer protocol, socket interface, client-server programming.

**Text Books:**

1. Computer Networks: A Systems Approach (Fifth Edition) by L. L. Peterson and B. S. Davie, publisher: Morgan Kaufmann.
2. Computer Networking: A Top-Down Approach (Fifth Edition) by J. F. Kurose and K. W. Ross, publisher: Pearson B.P. Lathi and Z. Ding, "Modern Digital and Analog Communication Systems," 4th Ed., Oxford University Press, 2009.

**Reference Books:**

1. TCP/IP Illustrated Volume 1: The Protocols, Second edition, K. R. Fall and W. Richard Stevens, publisher: Pearson.
2. B. A. Forouzan, "Data Communications and Networking," 4th Ed., McGraw Hill, 2012.

Course Code	Course Name	L-T-P: C
EC361	Computer Networks Lab	0-0-2: 1

**Objectives:** This course makes students aware of various cabling technologies used in different types of networks. Give exposure to various network commands. To implement various network and transport level protocols. Give exposure to Wireshark and packet tracer to simulate different types of networks.

**Learning Outcomes:**

1. Describe the functions of common networking devices and their role in data communication.
2. Implement various networking commands.
3. Implement various networking protocols such as flow and error control.
4. Have a working knowledge of socket programming.
5. Have a working knowledge of Wireshark and Cisco Packet Tracer.

**List of Experimental Tasks:**

1. Introduction to IIITV Network structure and basic Networking Hardware devices such as Hub, Switch, Router, Firewall, NIC, Modem, Gateway, etc.

2. Construction of different types of Networking Cables and their usage.
3. Usage of basic Networking Commands such as Ping, Ipconfig, Traceroute, Arp, Netstat, Whois, SSH, etc.
4. Error detection and correction.
5. Usage of Networking files in Ubuntu.
6. Socket Programming.
7. Flow Control-sliding window and go-back-n protocols.
8. Hands-on with Wireshark.
9. Hands-on with Cisco Packet Tracer.

**Text Books:**

1. Computer Networks: A Systems Approach (Fifth Edition) by L. L. Peterson and B. S. Davie, publisher: Morgan Kaufmann.
2. Computer Networking: A Top-Down Approach (Fifth Edition) by J. F. Kurose and K. W. Ross, publisher: Pearson B.P. Lathi and Z. Ding, "Modern Digital and Analog Communication Systems," 4th Ed., Oxford University Press, 2009.

**Reference Books:**

1. TCP/IP Illustrated Volume 1: The Protocols, Second edition, K. R. Fall and W. Richard Stevens, publisher: Pearson.
2. B. A. Forouzan, "Data Communications and Networking," 4th Ed., McGraw Hill, 2012.

Course Code	Course Name	L-T-P: C
EC303	Digital Signal and Image Processing	3-0-0: 3

**Objectives:** The course covers essential DSP topics—sampling, quantization, transform methods, digital filter design, multi-rate processing, adaptive filtering—and integrates them with digital image processing topics such as image formation, enhancement, restoration, compression, segmentation, and advanced applications. This integrated approach prepares students to design, analyze, and implement algorithms for communications, multimedia, and computer vision.

**Learning Outcomes:** On successful completion of this course, students should be able to:

1. Apply advanced transform methods (DFT, FFT, DCT, wavelets) for signal and image analysis.
2. Enhance and restore images using spatial and frequency-domain methods.
3. Develop image compression schemes and understand coding standards (e.g., JPEG/MPEG).

4. Extract image features using morphological operations and segmentation techniques.

**Prerequisite:** Signals and Systems, Linear Algebra

**Contents:**

**Module-I: Digital Signal Processing Essentials:** Overview of discrete time signal analysis and DSP applications, Sampling of continuous time signals (low pass and band pass), Uniform and non uniform quantization, adaptive quantization, Discrete Fourier Transform (DFT) properties and applications, FFT algorithms, Introduction to Discrete Cosine Transform (DCT)

**Module-II: Digital Image Fundamentals and Acquisition:** Image representation: pixels, resolution, and color models (RGB, HSI, YCbCr), Image sensing and acquisition; imaging devices and sensor characteristics, Image sampling and, quantization process

**Module-III: Image Processing Techniques:** Gray-level transformations and histogram processing, Spatial filtering: smoothing (mean, median) and sharpening filters, Frequency domain filtering using FFT: low pass, high pass, and notch filters, Image restoration: inverse filtering and Wiener filtering, Color image processing techniques

**Module-IV: Image Compression and Information Extraction:** Image Compression Models, Error-Free Compression, Lossy Compression, JPEG, MPEG. Wavelet transforms and multiresolution analysis for compression, basic entropy coding (Huffman coding), Morphological operations: dilation, erosion, opening, and closing, Image segmentation techniques: thresholding, clustering, and region growing

**Text Books:**

1. Gonzalez, R.C. & Woods, R.E., Digital Image Processing, 3rd Edition, Pearson, 2008.
2. Oppenheim, A.V. & Schaffer, R.W., Discrete Time Signal Processing, 3rd Edition, Prentice Hall, 2010.

**Reference Books:**

1. Jain, A.K., Fundamentals of Digital Image Processing, Prentice Hall, 2009.
2. Pratt, W.K., Digital Image Processing, John Wiley & Sons, 2006.
3. Proakis, J.G. & Manolakis, D.G., Digital Signal Processing: Principles, Algorithms, and Applications, 4th Edition, Pearson, 2007.

Course Code	Course Name	L-T-P: C
-------------	-------------	----------

EC363	Digital Signal and Image Processing Lab	0-0-2: 1
-------	---	----------

**Objectives:** To provide hands-on experience in implementing and simulating digital signal and image processing algorithms using MATLAB and Python. Lab exercises reinforce theory by enabling students to simulate, analyze, and process signals and images through practical experiments.

**Laboratory Outcomes:** On successful completion of this course, students should be able to:

1. Simulate sampling and quantization of continuous signals and images.
2. Implement Fourier transform algorithms (DFT, FFT) and compare FFT implementations.
3. Design and simulate FIR and IIR digital filters and perform multi-rate processing.
4. Acquire, display, and manipulate digital images and perform color space conversions.
5. Apply spatial and frequency-domain filtering to enhance and restore images.
6. Develop a basic JPEG-like compression scheme using transform coding techniques.
7. Execute segmentation and morphological operations for feature extraction.

**List of Experiments:**

1. Sampling and Quantization: Simulate sampling of continuous-time signals and quantization (uniform, non-uniform, and companding techniques).
2. Fourier Transform Implementation: Compute and analyze the DFT and FFT of signals; compare decimation in time and frequency methods.
3. Image Acquisition and Display: Read, display, and manipulate digital.
4. Enhance the contrast of a grayscale image by redistributing its intensity values using histogram equalization.
5. Spatial Domain Filtering: Apply intensity transformations, smoothing (mean, median), and sharpening filters to images.
6. Frequency Domain Filtering: Design and implement frequency-domain filters (low-pass, high-pass, notch) using FFT-based techniques.
7. Image Restoration: Implement inverse filtering and Wiener filtering to restore degraded images.
8. Image Compression: Develop a basic JPEG-like compression scheme using the DCT, with quantization and entropy coding (Huffman coding).
9. Image Segmentation and Morphological Processing: Apply thresholding, clustering, and morphological operations (dilation, erosion, opening, closing) for segmentation and feature extraction.
10. To implement the Hough Transformation.

**Text Books:**

1. Gonzalez, R.C. & Woods, R.E., Digital Image Processing, 3rd Edition, Pearson, 2008.
2. Gonzalez, R.C. & Woods, R.E., Digital Image Processing using Matlab, Pearson, 2004.
3. Oppenheim, A.V. & Schaffer, R.W., Discrete-Time Signal Processing, 3rd Edition, Prentice Hall, 2010.

**Reference Books:**

1. Jain, A.K., Fundamentals of Digital Image Processing, Prentice Hall, 2009

Course Code	Course Name	L-T-P: C
EC305	Digital Communication	3-0-0: 3

**Objectives:** To introduce the principles and techniques of digital communication systems, including signal representation, coding, modulation, and detection. The course provides an understanding of digital transmission, error control, and performance analysis in noise and bandwidth constraints.

**Learning Outcomes:**

1. Understand the fundamentals of digital communication and the differences from Analog systems.
2. Analyze and design digital modulation techniques such as ASK, PSK, and QAM.
3. Apply source and channel coding techniques for efficient transmission.
4. Understand inter-symbol interference (ISI) and design pulse shaping techniques.
5. Perform error detection and correction using various coding methods.
6. Evaluate the performance of digital communication systems in terms of BER & SNR.

**Prerequisite:** - Analog Communication, Signal & System

**Contents:**

**Module 1 - Introduction to Digital Communication:**

Elements of a digital communication system, advantages over analog communication; Sampling theorem and reconstruction of signals; Pulse code modulation (PCM), quantization, quantization noise; Differential PCM (DPCM), Delta modulation, adaptive delta modulation.

**Module 2- Digital Modulation Techniques:**

Amplitude shift keying (ASK), Frequency shift keying (FSK), Phase shift keying (PSK); Quadrature amplitude modulation (QAM), Differential PSK (DPSK);

Comparison of digital modulation techniques based on bandwidth, power, and noise immunity.

**Module 3- Transmission and Reception in Digital Communication:**

Baseband transmission: Inter-symbol interference (ISI), Nyquist criterion; Pulse shaping techniques and matched filter detection; Eye diagrams and equalization techniques for ISI mitigation; Coherent and non-coherent detection techniques.

**Module 4 - Information Theory and Coding:**

Introduction to information theory, entropy, and source coding; Huffman coding, Shannon's theorem, and channel capacity; Error control coding: Linear block codes, cyclic redundancy check (CRC), convolutional codes; Viterbi decoding and forward error correction techniques.

**Module 5 - Performance Analysis of Digital Communication Systems:**

Bit error rate (BER), signal-to-noise ratio (SNR); Spread spectrum techniques: Direct sequence and frequency hopping spread spectrum; Multiple access techniques: TDMA, FDMA, CDMA.

**Text Books:**

1. S. Haykin, *Digital Communications*, John Wiley & Sons, 2001.
2. B. P. Lathi, *Modern Digital and Analog Communication Systems*, 4th Ed., Oxford University Press, 2009.

**Reference Books:**

1. J. G. Proakis & M. Salehi, *Digital Communications*, 5th Ed., McGraw-Hill, 2007.

Course Code	Course Name	L-T-P: C
EC365	Digital Communication Lab	0-0-2: 1

**Objectives:** To provide hands-on experience with the simulation and implementation of digital communication techniques—including modulation, coding, pulse shaping, and performance analysis—using modern software tools and laboratory equipment.

**Laboratory Outcomes:**

1. The students will be able to design and simulate end-to-end digital communication systems.
2. The students will be able to implement and analyze various digital modulation and demodulation schemes.

3. The students will be able to evaluate system performance through Bit Error Rate (BER) and Signal-to-Noise Ratio (SNR) measurements.
4. The students will be able to apply source and channel coding techniques for error detection and correction.
5. The students will be able to implement pulse shaping and matched filtering techniques to mitigate inter-symbol interference (ISI).
6. The students will gain proficiency in using simulation tools (e.g., MATLAB/Simulink) for digital communication experiments.

#### List of Experiments

1. **PCM and Delta Modulation:** Simulation of sampling, quantization, PCM encoding/decoding, and delta modulation.
2. **Digital Modulation Schemes:** Implementation and analysis of ASK, FSK, and PSK techniques.
3. **Quadrature Amplitude Modulation (QAM):** Simulation and performance evaluation of QAM under varying channel conditions.
4. **Baseband Transmission and Pulse Shaping:** Design of pulse shaping filters, matched filtering, and analysis of ISI using eye diagrams.
5. **Error Control Coding – Part I:** Implementation of Hamming codes and cyclic redundancy check (CRC) for error detection.
6. **Error Control Coding – Part II:** Simulation of convolutional coding and Viterbi decoding techniques.
7. **BER Performance Analysis:** Measurement of Bit Error Rate (BER) in digital systems under AWGN channel conditions.
8. **Spread Spectrum Techniques:** Simulation of Direct Sequence Spread Spectrum (DSSS) and Frequency Hopping Spread Spectrum (FHSS).
9. **Multiple Access Techniques:** Simulation of TDMA, FDMA, and CDMA methods in digital communication systems.
10. **Software-Defined Radio (Advanced Experiment):** Implementation of a basic digital communication system using an SDR platform.

#### Text Books:

1. S. Haykin, *Digital Communications*, John Wiley & Sons, 2001.
2. B. P. Lathi, *Modern Digital and Analog Communication Systems*, 4th Ed., Oxford University Press, 2009.

#### Reference Books:

1. J. G. Proakis & M. Salehi, *Digital Communications*, 5th Ed., McGraw-Hill, 2007.

Course Code	Course Name	L-T-P: C
EC367	Microprocessor and Microcontroller Lab	0-1-2: 2

**Objectives:** The objective of a Microprocessor course is to provide students with a comprehensive understanding of microprocessor architecture, programming, and applications. The course aims to familiarize students with the internal structure of microprocessors, including the ALU, registers, control units, and memory organization. The objective of a Microcontroller course is to equip students with a solid understanding of microcontroller architecture, programming, and its practical applications in embedded systems. The course aims to provide in-depth knowledge of microcontroller components such as the central processing unit (CPU), memory, input/output ports, timers, and communication interfaces.

**Prerequisite:** Digital logic design, fundamentals of c/assembly/Verilog/VHDL programming, basics of computer

#### Laboratory Outcomes:

1. Understanding of Microprocessor Architecture
2. Assembly Language Programming
3. Interfacing
4. Hands-on Experience with Development Tools
5. Interrupts and Timing Mechanisms
6. Microprocessor-based System Design
7. Troubleshooting and Debugging
8. Communication Protocols
9. Application Development

#### List of Experiments:

1. Introduction of 8085 Microprocessor and Its Trainer kit.
2. Performing assembly language Programming of addition,
3. Subtraction, multiplication, division using 8085 Microprocessors.
4. To perform sorting of 10 numbers in 8085 microprocessor.
5. To find the factorial of a given number and searching the smallest number in a given array in 8085 microprocessors. Find the sum of a series of numbers in 8085 microprocessor.
6. Design an 8051  $\mu\text{C}$  based circuit and its associated program to implement 4 bit binary counter using LEDs.
7. Design an 8051  $\mu\text{C}$  based circuit and its associated program to implement a 4 bit binary up/down counter using LEDs with the stated specifications.

8. Design a 8051  $\mu\text{C}$  based circuit and its associated program to interface two seven segment displays.
9. Design a circuit associated program interface a DAC with 8051 microcontroller and generate a ramp output through it.
10. Interface an ADC with 8051 Microcontroller and Show the Corresponding Output in Two Seven Segment Displays

**Text Books:**

1. R. S. Gaonkar, *The 8085 Microprocessor: Architecture, Programming, and Interfacing*, 5th ed., New Jersey: Prentice-Hall, 2002.
2. Mazidi, Muhammad Ali, Janice Gillispie Mazidi, and Rolin D. McKinlay. 2006. *The 8051 Microcontroller and Embedded Systems*. 2nd ed. Pearson Prentice Hall.

Course Code	Course Name	L-T-P: C
EC307	Introduction To VLSI Design	3-0-2: 4

**Objective:** The primary objective of the "Introduction to VLSI Design" course is to provide students with a foundational understanding of the principles and techniques used in the design and development of Very-Large-Scale Integration (VLSI) systems. The course aims to cover the various aspects of VLSI design, including digital logic design, CMOS technology, fabrication processes, design methodologies, and the tools used for the design and simulation of integrated circuits. By the end of the course, students should be equipped with the knowledge and skills necessary to approach VLSI system design with a solid understanding of both theoretical and practical aspects.

**Learning Outcomes:**

1. Understand the fundamental principles of VLSI technology and its impact on modern electronics.
2. Gain knowledge of CMOS technology, including MOSFET operation and CMOS logic gates.
3. Learn the VLSI design flow, from specification and design entry to simulation, synthesis, layout, and verification.
4. Design and analyze combinational and sequential digital circuits using VLSI techniques.
5. Understand the physical design process, including layout, floor planning, and routing in VLSI systems.
6. Apply design automation tools (EDA tools) for logic synthesis, timing analysis, and verification.
7. Evaluate power, performance, and area (PPA) trade-offs in digital circuit designs.

8. Gain hands-on experience with industry-standard tools for VLSI design, simulation, and verification.
9. Develop an understanding of the concepts behind FPGA and ASIC design for hardware implementation.

**Prerequisite:** Digital logic design, Electronic Devices and Circuits, fundamentals programming, basics of computer architecture

**Contents:**

**Module-I:** Overview of VLSI technology: Evolution, trends, and applications of VLSI, VLSI Design Flow: Overview of the steps involved in VLSI design (specification, design entry, synthesis, verification, and testing). Design styles: Full-custom, semi-custom, and programmable logic. Basic properties of semiconductors, MOSFETs, Threshold voltage, drain current, subthreshold conduction.

**Module-II:** VLSI Fabrication Technology: Wafer Fabrication and Lithography: Overview of semiconductor wafer fabrication, photolithography techniques, Etching, Diffusion, and Deposition: Processes involved in creating MOS transistors, Packaging and Testing: Introduction to the packaging of VLSI chips and testability techniques. CMOS Technology, CMOS Transistor and Circuit Design: Operation, parameters, and characteristics of MOSFETs. CMOS Process Technology: Fabrication steps, materials used, and layout design rules. Scaling and Moore's Law: Impact of scaling on device performance, power, and area.

**Module-III:** Basic Digital Circuits: Combinational and sequential logic circuits, Gate-Level Design: Implementation of logic gates and basic digital modules using CMOS. Designing Arithmetic Circuits: Adders, multipliers, Basic CMOS Inverter Design: Design and analysis of the CMOS inverter, Speed-Power Trade-off: Analysis of delay, power consumption, and performance optimization in logic circuits, Static and Dynamic Logic Design: Analysis of static CMOS logic and dynamic logic circuits, transistor sizing.

**Module-IV:** Power and Area Optimization, Low Power Design, Techniques for reducing power consumption in CMOS circuits (e.g., clock gating, power gating, voltage scaling). Area Efficiency: Design strategies to minimize chip area while maintaining functionality.

**Module-V:** Test Generation: Methods for generating test vectors to detect faults. Built-in Self Test (BIST): Fault Models: Stuck-at faults. VLSI Testing and Recent Trends and Future Directions in VLSI: Advanced CMOS Technologies: FinFETs, SOI, and other emerging transistor technologies. 3D ICs and Integration: Quantum Computing and Nanoelectronics:

Future possibilities in VLSI beyond traditional CMOS scaling.

**Text Books:**

1. CMOS Digital Integrated Circuits: Analysis and Design" by Sung-Mo (Steve) Kang and Yusuf Leblebici.
2. Digital Integrated Circuits: A Design Perspective" by Jan M. Rabaey.
3. VLSI Design" by Douglas A. Pucknell and Kamran Eshraghian.

**Reference Books:**

1. Principles of CMOS VLSI Design: A Systems Perspective" by Neil Weste and Kamran Eshraghian.
2. VLSI Design and EDA Tools" by Michael J. Flynn.

Course Code	Course Name	L-T-P: C
EC369	Microwave Engineering Lab	0-1-2: 2

**Objectives:** To introduce the students to various microwave sources, components, and equipment and the measurement of their performance characteristics.

**Learning Outcomes:**

1. Understand the fundamental principles of microwave transmission, propagation, and scattering.
2. Gain hands-on experience with microwave components such as waveguides, antennas, and resonators.
3. Measure and analyze microwave signal power, frequency, and impedance.
4. Investigate microwave measurements, including S-parameters, and interpret the results.
5. Develop skills in designing and building simple microwave circuits and systems.
6. Understand the characteristics of different types of microwave sources, detectors, and modulators.
7. Analyze microwave signals in different media (e.g., air, dielectric materials, and waveguides).
8. Learn the applications of microwaves in communication, radar, and medical technologies.

**List of Experiments:**

1. Study of microwave sources and components.
2. Study of crystal detector characteristics.
3. Measurement of VSWR, impedance and frequency.
4. Measurement of attenuation and dielectric constant.
5. Measurement of phase shift.

6. Measurement of Q of a cavity resonator.
7. Measurement of directional coupler characteristics.
8. Study of tee junctions.
9. Study and measurement of transmission line characteristics.
10. Measurement of antenna characteristics.
11. Study of Spectrum Analyzer.
12. Study of Network Analyzer.

**Text/Reference Books:**

1. Pozar, D.M., "Microwave Engineering", 3rd Ed., John Wiley & Sons,2004.
2. Collin, R.E., "Foundations for Microwave Engineering", 2nd Ed., John Wiley & Sons, 2000.
3. Laverghetta, T.S., "Microwave Measurements and Techniques", Artech House,1984.
4. Laverghetta, T.S., "Practical Microwaves", H.W. Sams & Co,1984.

Course Code	Course Name	L-T-P: C
IT305	Information Security	3-0-0: 3

**Objectives:** The course intends to provide the basic foundations of information security and its impact on the IT infrastructure. The course includes not only the cryptographic techniques used in information security, but it also focuses on the hands-on experience with tools and techniques used to ensure network security.

**Learning outcomes:** On successful completion of this course, students should be able to:

1. List the importance of information security of IT infrastructure.
2. Use the cryptographic algorithms to ensure the confidentiality, integrity, availability, etc. of information.
3. Analyze the security strengths of cryptographic algorithms.
4. Learn the network security protocols used to ensure the web application security.
5. Identify network security threats and the means to prevent attacks through tools/techniques such as firewalls, PGP, etc.

**Prerequisites:** MA102 and IT101.

**Contents:**

**Module I: Symmetric-key Cryptography:** Classical ciphers, Feistel structure, Modern block ciphers such as Data Encryption Standard (DES) and Advanced Encryption Standard (AES), Cryptographic hash functions, Message authentication codes (MAC).

**Module II: Asymmetric-key Cryptography:**

Applications of asymmetric-key cryptosystem, RSA cryptosystem, Diffie-Hellman key exchange protocol, Digital signature.

**Module III: User Authentication and Access Control:**

Password- Based/Token-Based/Biometric Authentication, Remote User Authentication, Security Issues for User Authentication, Access Control Principles, Subjects, Objects, and Access Rights, Linux/UNIX Filesystem Access Control, Role-Based/Attribute-Based Access Control, Identity, Credential, and Access Management.

**Module IV: Network Security:**

Secure socket layer (SSL)/ Transport layer security (TLS), Firewalls, Packet filtering firewall, Stateful inspection firewalls, E-Mail security, Pretty good privacy (PGP), Secure Multipurpose Internet Mail Extensions (S/MIME), IP security overview, IP security policy, Encapsulating security payload (ESP).

**Module V: Management Issues:**

Security Risk Assessment, Security Risk Analysis, Security Plan, Monitoring Risks, Security Awareness/Training/Education, Security policies, Standards, Ethics.

**Text Books:**

1. Stallings and Brown, *Computer Security: Principles and Practice*, Third Edition, Publisher: Pearson, 2014.
2. M. Bishop, *Computer Security: Art and Science*, Reprint Edition, Addison-Wesley Professional, 2015.
3. Kaufman et al., *Network Security*, Second Edition, Publisher: Prentice Hall, 2002.

**Reference Books:**

1. Trappe and Washington, *Introduction to Cryptography with Coding Theory*, Second Edition, Publisher: Pearson, 2009.
2. Bruce Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C*, Second Edition, Publisher: Wiley, 1996.

Course Code	Course Name	L-T-P: C
IT365	Information Security Lab	0-0-2: 1

**Objectives:** The laboratory course intends to provide a platform to implement the state-of-the-art cryptographic algorithms, detect and prevent security vulnerabilities in the IT infrastructure and effectively use and apply the state-of-the-art tools/techniques to ensure information security.

**Learning outcomes:** On successful completion of this course, students should be able to:

1. Detect the vulnerabilities in the IT infrastructure.
2. Implement the state-of-the-art cryptographic algorithms.
3. Effectively use and apply the tools/techniques such as Pretty Good Privacy, Firewall, etc. to ensure the information security.
4. Effectively use and apply the tools/techniques such as Wireshark, Nmap (Network Mapper), Metasploit, etc. to ensure the information security.

**Prerequisites:** IT101, IT161, MA102, CS203 and CS263.

**Contents:**

Laboratory assignments will consist of following topics to develop creativity and analytical abilities of students. The assignments should be in the form of hands-on experience of cryptosystems, in one (or more) of the programming language(s), and network security protocols.

1. Classical ciphers to understand encryption, decryption, cryptanalysis.
2. Data Encryption Standard.
3. Advanced Encryption Standard.
4. Cryptographic hash algorithm such as SHA-512.
5. Message authentication codes such as HMAC.
6. RSA cryptosystem, Diffie-Hellman key exchange, Elgamal cryptosystem (1024/2048-bits).
7. Secure socket layers (SSL)/Transport layer security (TLS).
8. Firewalls (e.g. iptables).
9. Pretty good privacy (PGP).
10. Wireshark.
11. Nmap (Network Mapper), Metasploit

**Reference:** Laboratory instructions and handouts.

Course Code	Course Name	L-T-P: C
IT307	Software Project Management	2-0-0: 2

**Course Objectives:** The objective of the course is to let the student to understand the project management framework. Learn about the project phases, project life cycle, project stakeholders, and key general management skills. Learn about the supporting project management knowledge areas like quality and human resources.

**Learning outcomes:** Skill set for an entry level software project manager/ team lead in the software industry.

**Prerequisite:** CS303.

**Contents:**

**Introduction:** Introduction to project management; Open source tools, merits and limitations.

**Software Measurement:** software metrics, cyclomatic complexity, class cohesion metrics.

**Software Estimation:** Cost estimation, effort estimation, schedule estimation, duration estimation.

**Software Management:** Software planning; configuration management; software tendering and contracting processes; risk management.

**Project Execution and Quality:** Project execution; quality insurance, deadline management, configuration management.

**Standards and Methodologies:** RFPs, IETF, ISO, IEEE standards.

**Web Based Open Source Project Management Tools:** Simulation/emulation, performance measures, applications.

**Text Book:**

1. *Software Project Management: A Process-Driven Approach* -- A. Ahmed: Auerbach Publications, 2011

**Reference Books:**

1. *Applied Software Project Management* -- A. Stellman and J. Greene: O'Reilly Media, 2005.

Course Code	Course Name	L-T-P: C
IT367	Software Project Management Lab	0-1-2: 2

**Course Objectives:** The objective of the laboratory course is let the students to learn about the phases of software projects, project life cycle, project stakeholders, and general management skills.

**Learning Outcomes:** Skill set for an entry level software project manager/ team lead in the software industry.

**Prerequisites:** IT101 and IT161.

**Contents:**

Laboratory sessions based on understanding project management concepts, project planning, project monitoring, risk management, resource management, project metrics analysis, subcontract management,

quality management and models, tools and frameworks for project management.

**Reference:** Laboratory instructions and handouts.

Course Code	Course Name	L-T-P: C
IT309	Management Information System	3-0-0: 3

**Course Objectives:** This course provides a foundation in the theory and practical application of information systems within an organization. The course includes strategic value, methodologies, quality, decision making, business-driven information systems.

**Learning outcomes:** On successful completion of this course, students should be able to:

1. Analyze, evaluate and make recommendations regarding business technology and decisions.
2. Have problem solver approach using critical thinking and make informed decision.
3. Work in team collaboratively, demonstrating courtesy and etiquettes.

**Prerequisite:** CS303.

**Contents:**

**Introduction:** Technology of Information Systems, concepts, definition; role and impact of MIS; role and importance of management; approaches to management; functions of the manager; management as a control system; concepts of data models; database design; client- server architecture.

**Process of Management:** Planning, organization, staffing, coordination and controlling; management by exception; MIS as a support to management; organization structure and theory; basic model and organization structure; organizational behavior.

**Decision Making and Information:** Decision making concepts, methods, tools and procedures; behavioral concepts in decision making; organizational decision making; information concepts as a quality product; classification of the information; methods of data and information collection; value of the information; organization and information system concepts, control types; handling system complexity; post implementation problems in systems.

**System Analysis and Design:** Need for system analysis; system analysis of existing system; new requirement; system development model; structured system analysis and design; computer system design; development of MIS; development of long range plans

of the MIS; ascertaining the class of the information; determining the information requirement; development and implementation of the MIS; management of quality; MIS factors of success and failure.

**Decision Support Systems:** Deterministic systems; artificial intelligence; knowledge based systems; MIS and the role of DSS; enterprise management systems; enterprise resource planning (ERP); ERP features and benefits; implementation factors of ERP; Internet and Web based information system; Electronic Commerce.

**Text Book:**

1. *Management Information Systems*, K. C Landon, J.P. Laudon, Prentice Hall, 2000.

**Reference Book:**

1. *Management Information Systems*, G. B. Davis, M. H. Olson, McGraw Hill, 1998.

Course Code	Course Name	L-T-P: C
MC301	Advanced Statistics	3-1-0: 4

**Objectives:** The objective of this course is to develop a strong theoretical and practical understanding of statistical inference for estimation and detection in stochastic systems. The course focuses on constructing mathematical models for observations corrupted by randomness, designing optimal estimators and detectors using principled criteria (e.g., MVUE, CRLB, ML/MAP, Neyman-Pearson), and applying these methods to real-world problems such as time-series modeling and Kalman filtering.

**Learning Outcome:** By the end of this course, students will be able to:

1. Formulate statistical inference problems involving estimation and detection using stochastic process models.
2. Apply hypothesis testing frameworks, including the Neyman-Pearson theorem, to derive optimal detectors and analyze false alarm and detection probabilities.
3. Derive and interpret Minimum Variance Unbiased Estimators (MVUE) using criteria such as sufficiency and completeness.
4. Compute and use the Cramér-Rao Lower Bound (CRLB) to assess the performance limits of unbiased estimators for scalar and vector parameters, including applications to Gaussian random processes.

5. Implement and interpret Maximum Likelihood (ML) and Maximum A Posteriori (MAP) estimation techniques for parameter estimation.
6. Model and analyze stochastic processes and time series, and apply appropriate filtering techniques.
7. Apply Kalman filtering for recursive state estimation in dynamic systems.
8. Evaluate detector performance using metrics like probability of error, ROC curves, and likelihood ratio rules.
9. Connect theory to applications in communications, signal processing, data science, and machine learning.

**Prerequisite:** Probability and Statistics.

**Contents:**

Review of hypothesis testing and random processes, Mathematical problem of Estimation, Assessing Estimator Performance.

Minimum Variance Unbiased Estimation (MVUE): unbiased estimators, minimum variance criterion, existence of MVUE, finding MVUE.

Cramer-Rao lower bound (CRLB) and Sufficient Statistics: definition, general CRLB for transformations, vector parameter, asymptotic CRLB for Gaussian Random Processes.

Maximum likelihood estimation, Maximum A Posterior probability estimation

Kalmen filtering, Time series analysis.

Detection theory: Likelihood Ratio Test (LRT), Neyman Pearson Theorem, Receiver Operating Characteristics (ROC) & Performance Measures, Bayesian Detection & Risk, Random signals with unknown parameters, summary of detectors.

**Textbook:**

1. *Fundamentals of Statistical Processing, Volume 1: Estimation Theory Vol I & II*. Steven M. Kay, Pearson Education.

**Reference book:**

1. *Mathematical Methods and Algorithms for Signal Processing*. Todd K. Moon and Wynn C. Stirling, Prentice Hall.

Course Code	Course Name	L-T-P: C
MC303	Graph Theory	3-0-0: 3

MC303: Graph Theory

**Objectives:** This course introduces the fundamental principles of graph theory structures, and algorithms of graph theory. This course has applications in network analysis, reinforcement learning, and combinatorial optimization. Also, introduce connections between graph theory and algebraic structures, particularly edge ideals in commutative algebra.

**Learning Outcomes:** After successful completion, students will be able to:

1. To analyze and model real-world systems using graph-theoretic concepts and algorithms.
2. Apply and analyze graph algorithms including shortest path algorithms and greedy algorithms for minimum spanning trees.
3. Understand advanced topics such as Ramsey numbers and algebraic graph theory, including eigenvalues, spectra, and graph polynomials.

**Prerequisite:** Discrete Mathematics, Linear Algebra,

**Contents:**

**Review:** Graphs, subgraphs, isomorphism, representation of graphs, degrees and graphical sequences, walks, trails, paths, cycles, Connectivity, bipartite graphs.

**Trees and connectivity:** Characterizations of trees, Properties of trees, minimum-spanning-trees, number of trees, Cayley's formula, shortest path algorithms, cut-sets, Characterization of blocks, Menger's theorem, Vertex Connectivity and Edge Connectivity. Minimum spanning tree problem: greedy algorithm. Kruskal's and Prime's algorithms.

**Eulerian and Hamiltonian graphs:** Characterizations, Necessary/sufficient conditions. Basic relations, Tutte's Perfectmatching theorem and consequences.

**Graph coloring:** Edge-colorings of bipartite graphs, Gupta Vizing's theorem, greedy algorithm for vertex-colorings, Brook's theorem, clique-number and vertex chromatic number.

**Planar graphs:** Euler's formula and its consequences, Kuratowski's Characterization.

**Directed graphs:** Basics, various connectivities and tournaments.

**Matching:** Matchings in bipartite graphs, Vertex-cover problem, Max Cliques problem, Independent set, Hall Theorem, Related algorithm questions,, Maximum matching number.

**Ramsey Theory:** Classical Ramsey Number, Exact Ramsey Number and bounds, Graph Ramsey Theory.

**Algebraic Graph Theory:** Operations on Graphs and the Resulting Spectra: the polynomial of a graph, eigenvalues and eigenvectors, line graphs and total graphs.

**Textbooks:**

1. J.A Bondy and U.S.R Murthy, Graph Theory with Applications, Macmillan, 1976.
2. Combinatorics and Graph Theory, John M. Harris • Jeffrey L. Hirst • Michael J. Mossinghoff, 2008.

**References:**

1. Douglas B. West, "Introduction to Graph Theory", Prentice Hall (2001).
2. Biggs N. *Algebraic Graph Theory*, Cambridge University Press.

Course Code	Course Name	L-T-P: C
MC305	Introduction to Modern Algebra & Cryptography	3-0-0: 3

**Objectives:** The course intends to provide a theoretical knowledge and hands-on experience of cryptographic algorithms, cryptanalysis, and network security protocols, used to ensure the security of data.

**Learning Outcomes:** On successful completion of this course, students should be able to:

1. Understand the role of cryptography and cryptanalysis in security.
2. Effectively use the cryptographic algorithms to ensure the confidentiality, integrity, availability, etc. of information.
3. Effectively analyze the security strengths/weaknesses of cryptographic algorithms.
4. Learn the network security protocols used to ensure the web application security.

**Prerequisites:** Discrete Mathematics.

**Contents:**

**Secret Key Cryptography:** Substitution-Permutation network, Feistel structure, Block ciphers, Data encryption standard (DES), Advanced encryption standard (AES), Stream ciphers.

**Modes of Operation:** Electronic code book mode(ECB), Cipher block chaining mode (CBC), Cipher feedback mode (CFB), Output feedback mode (OFB), Counter mode (CTR).

**Concepts of Number Theory and Finite Fields:** Euclidian algorithm, Modular arithmetic, Groups, Rings, Finite fields, Polynomial arithmetic, Fermat's

theorem, Euler's theorem, Chinese remainder theorem (CRT), Integer factorization problem, Discrete logarithm problem, Elliptic curve discrete logarithm problem (ECDLP).

**Data Integrity:** Cryptographic Hash functions, Secure hash algorithm (SHA2 or SHA3); Message authentication codes (MAC).

**Public Key Cryptography:** RSA cryptosystem, Rabin's cryptosystem, Diffie-Hellman key exchange, Elgamal cryptosystem, Digital signature algorithm (DSA), Elliptic curve arithmetic, Elliptic curve digital signature algorithm (ECDSA).

**Security Models:** Ciphertext-only attacks, Known-plaintext attacks, Chosen plaintext attacks (CPA), Chosen ciphertext Attacks (CCA), Adaptive chosen ciphertext attacks (CCA2).

**Modern Algebra:** Groups, subgroups, normal subgroups, quotient groups, homomorphisms; cyclic groups, permutation groups, Group action, Sylow's theorems and their applications; Rings, ideals, prime and maximal ideals, quotient rings, unique factorization domains, Principle ideal domains, Euclidean domains, polynomial rings, Eisenstein's irreducibility criterion; Fields, finite fields, field extensions, algebraic extensions, algebraically closed fields.

#### Text Books:

1. William Stallings, Cryptography and Network Security - Principles and Practice, Seventh Edition, Publisher: Pearson, 2016.
2. Dummitt, D.S. and Foote, R.M. Abstract Algebra. 3rd Edition, John Wiley & Sons, Inc., 2004.
3. Trappe and Washington, Introduction to Cryptography with Coding Theory, Second Edition, Publisher: Pearson, 2009.
4. Katz & Lindell, Introduction to Modern Cryptography: Principles and Protocols, Second Edition, Publisher: Chapman & Hall/CRC, 2014.
5. Gallian, Joseph. Contemporary Abstract Algebra. United States: CRC Press, 2025.

#### Reference Books:

1. Bruce Schneier, Applied Cryptography: Protocols, Algorithms, and Source Code in C, Second Edition, Publisher: Wiley, 1996.
2. Douglas R. Stinson, Cryptography: Theory and Practice, Third Edition, Publisher: Chapman and Hall/CRC, 2005.
3. Kaufman et al., Network Security, Second Edition, Publisher: Prentice Hall, 2002.

Course Code	Course Name	L-T-P: C
MC365	Introduction to Modern Algebra & Cryptography Lab	0-0-2: 1

**Objectives:** The Laboratory course is intended to Implement and empirically analyze the state-of-the-art cryptographic algorithms used for encryption, authentication, digital signatures, key exchange, etc. Empirically analyze the network security protocols such as HTTPs. Effectively use the tools and techniques such as Pretty Good Privacy, etc.

**Learning Outcomes:** On successful completion of this course, students should be able to:

1. Implement and analyze the classical ciphers, symmetric-key cryptosystems, and asymmetric-key cryptosystems.
2. Encrypt and decrypt the information using symmetric-key algorithms.
3. Encrypt and decrypt the information using asymmetric-key algorithms.
4. Protect the integrity of the information using symmetric-key and asymmetric-key algorithms.
5. Perform the secure key-exchange using asymmetric-key algorithms.
6. Digitally sign and verify the information.
7. Effectively use the network security tools.

**Prerequisites:** IT101, IT161, MA102, CS263, CS301 and CS361.

**Contents:** Laboratory assignments will consist of the following topics to develop creativity and analytical abilities of students. The assignments should be in the form of hands-on experience of cryptosystems, in one (or more) of the programming language(s), and network security protocols.

1. Classical ciphers to understand encryption, decryption, cryptanalysis.
2. Data Encryption Standard.
3. Advanced Encryption Standard.
4. Generation of S-Boxes used in AES.
5. Modes of operations and their analysis.
6. Cryptographic hash algorithm such as SHA-512.
7. Message authentication codes such as HMAC.
8. Number theory related concepts such as Chinese remainder theorem (CRT).
9. Analyze the impact of various parameter sizes on number theoretic problems such as integer factorization problem, Discrete logarithm problem, Elliptic curve discrete logarithm problem (ECDLP).

10. RSA cryptosystem, Diffie-Hellman key exchange, Elgamal cryptosystem (1024/2048-bits).
11. Implementation of elliptic curve arithmetic.
12. Elliptic curve digital signature algorithm (ECDSA).
13. Computational Group Theory related problem (Algorithm for Minimum Generating set of finite group and permutation group representation),
14. Secure socket layers (SSL)/Transport layer security (TLS).
15. Secure shell (SSH).
16. Firewalls (e.g. ip tables)

**Reference:** Lab Manual.

Course Code	Course Name	L-T-P: C
HS301	Entrepreneurship and IP Strategy	0-1-0: 1

#### Objectives:

1. To discuss intellectual property strategy to protect inventions and innovations of new ventures.
2. To develop skills of commercial appreciation by allocating knowledge about substantive aspects of management, strategy and legal literature.
3. The course will make participants appreciate the nature, scope and differences of IP, its different utilities and approaches.
4. The course will make participants to manage and strategize IP lifecycle effectively throughout the journey of start-up, in a time when it is aspired highly by the economy and society.
5. Participants will learn the fundamentals and advanced strategies of IP. They will be given opportunity for understanding the same in MSME sector. They will be finally be provided brief exposure about the valuation techniques and audits of IP.

#### Contents:

**Module I:** Introduction to entrepreneurship and intellectual property: Definition, Concepts, Introduction, What is an entrepreneurship? What do you understand by IP? Whether entrepreneurship and IP related? What is role of IP strategy in entrepreneurship? Case study I – IT industry.

**Module II:** Innovation and entrepreneurship: Innovation, invention and creativity; Types of innovation; Innovation, market and IP; Open innovation and IP; Case Study II - Biotechnology

**Module III:** IPR: Trademark and entrepreneurship: Trademark-Definations, Types, Registration, Infringement, Case study III - Textile industry

**Module IV:** IPR: Patent and entrepreneurship: Patent-introduction, infringement, strategies- I, strategies- II, Capsule version.

**Module V:** IPR: Copyright and entrepreneurship: Definition and subject matter, Copyright and related rights, Copyright registration and entrepreneurship, Copyright infringement, Case study IV – Film industry

**Module VI:** IPR: Industrial design and entrepreneurship: Industrial Design- Definition, concept, Industrial Designs Act - Key features, Industrial Design-Business, Industrial Design infringement, Case study V - Automobile industry

**Module VII:** IP strategy & entrepreneurship: IP strategy for start-up and MSME, IP transaction - introduction, IP valuation, bank loan, insurance, Success story and business model of a few start-ups, Case Study VI – Pharma industry and Agriculture

**Module VIII:** Entrepreneurship & IP - Government initiatives: Incubators, research parks, Various Government policies, Integrative approach – Entrepreneurship & IP strategy, Capsule revision, Am I ready to venture my start up? (Course applicability)

#### Books and References:

1. Ove Granstrand, The Economic and management of Intellectual Property, (1999)
2. Narayanan, V. K., Managing technology and innovation for competitive advantage, first edition, Pearson education, New Delhi, (2006)
3. Idris, K. (2003), Intellectual property: a power tool for economic growth, second edition, WIPO publication no. 888, Switzerland
4. Bosworth D. & Webster E , The Management of Intellectual Property, Edward Elgar.
5. Berman, Ideas to Assets, Wiley publications.
6. Richard Dorf & Thomas Byers, Technology ventures from idea to enterprise, 2nd edition.

#### Additional Reading:

WIPO - <http://www.wipo.int/patents/en/>

Course Code	Course Name	L-T-P: C
DP391	Design Project#	0-0-0: 3

*#The Design Project is a Pass/Fail course. It is conducted during summer vacation between 4<sup>th</sup> and 5<sup>th</sup> semester. The evaluation of project work is carried out in 1<sup>st</sup> week of 5<sup>th</sup> semester and its grade will appear in the 5<sup>th</sup> semester grade sheet.*

**Scope of the Project:** To provide an opportunity to second year undergraduate students to apply individual

efforts involving the knowledge of Information Technology, Computer Science, Embedded Systems or related areas in designing/ developing/ implementing complete systems.

Students are expected to carry out design project either individually or in a group of 2 – 4 students. The duration of the project work is between 6 – 8 weeks. Followings are the broad areas in which a student may work:

1. Software Development.
  2. Software Maintenance.
  3. Software Tools and Techniques.
  4. Hardware Development/Implementation.
  5. Robotics.
  6. Embedded System (Software & Hardware combine) Development/Implementation.
-

## Semester-VI

### Courses and Contents

Course Code	Course Name	L-T-P: C
CS302	Distributed and Parallel Computing	3-0-0: 3

**Description:** This course covers the foundations of distributed systems including models of computing, different types of communication (Layered Protocols, Remote Procedure Calls, Remote Objects, messages, streams), process models (threads, client/server, code migration and software agents), naming of entities, logical clocks and synchronization. The course will include programming assignments, project and project presentation. Specific language mastery is not important, though using one of C, C++, or Java will be essential.

**Learning Outcomes:** On successful completion of this course, students should be able to:

1. Learn architecture, algorithms and computational models used in Distributed, Cloud and parallel systems.
2. Understanding of various trade-offs involved in system design and performance analysis.
3. Understand various libraries for Distributed, Parallel and cloud computing platforms using various case studies.

**Prerequisites:** IT101, CS201, CS204 and CS301.

#### Contents:

**Module I:** Overview of C, UNIX and UNIX system calls. Introduction to distributed system, goals, hardware concepts, software concepts, the client-server model.

**Module II:** Communication: Layered Protocols, Remote Procedure Call, Remote Object Invocation, Message-Oriented and Stream-Oriented Communication.

**Module III:** Processes: Threads, Servers, Code Migration and Software Agents. Naming: Naming Entities, Locating Mobile Entities, Removing Unreferenced Entities. Synchronization: Clock Synchronization, Logical Clocks and Election Algorithms.

**Module IV: Parallel Computing:** Parallel Computer memory architectures, Programming Models, Shared memory model, Flynn's Programming model, Pipeline

computations. Automatic vs Manual Parallelization, Data dependencies, Load balancing.

**Module V: Cloud Computing:** Cloud Computing Models, Web Application Framework, Cloud Web Services, Service Oriented Architectures towards Cloud Computing.

#### Text Books:

1. *Distributed Systems: Principles and Paradigms*, Andrew Tanenbaum and Maarten van Steen, Prentice Hall.
2. *Distributed Systems: Concepts and Design*, George Coulouris, Jean Dollimore, and Tim Kindberg.

Course Code	Course Name	L-T-P: C
CS362	Introduction to Distributed and Parallel Computing Lab	0-0-2: 1

**Objectives:** The course aims to provide practical exposure of current computational paradigms. A student is expected to evaluate distributed, parallel and cloud architecture for various real life problems.

**Learning Outcomes:** On successful completion of this course, students should be able to:

1. Implement algorithms in distributed, cloud and parallel architecture.
2. Evaluate various models on performance and scalability trade off.
3. Design and implement real life application on different models of computation.

**Prerequisites:** CS101, CS201, CS204 and CS301.

#### Contents:

The lab part involve implementation of following problems:

1. Basic client-server in distributed environment.
2. Multi-client single server in distributed environment.
3. Multi-client multi server in distributed environment.
4. Java RMI.
5. Distributed Chat application.
6. Distributed Banking application.
7. Distributed File system application.
8. Matrix multiplication using parallel programming.
9. Real life problems on OpenMP and MPI libraries.
10. Working AWS and Google Colab architectures.
11. Performance analysis of various methods on GPU.

**Reference:** Laboratory instructions and handouts.

Course Code	Course Name	L-T-P: C
IT302	Cloud Systems Management	2-0-0: 2

**Course Objectives:** To introduce fundamental concepts of cloud computing and provide knowledge of managing cloud-based infrastructure, services, and resources.

**Learning Outcomes:** After completing this course, students will be able to:

1. Understand cloud service and deployment models.
2. Manage cloud resources and virtualized environments.
3. Monitor, secure, and optimize cloud systems.
4. Apply basic cloud management tools and practices

**Contents:**

**Module I:** Introduction to Cloud Computing- Overview of cloud computing, characteristics, service models (IaaS, PaaS, SaaS), deployment models (public, private, hybrid), virtualization concepts, hypervisors, and cloud architecture.

**Module II:** Cloud Infrastructure Management- Virtual machines, containers, storage management, networking in cloud, resource provisioning, scalability, load balancing, and basic cloud platforms overview.

**Module III:** Cloud Monitoring and Security- Cloud monitoring tools and techniques, logging, performance management, fault tolerance, identity and access management (IAM), data security, and compliance in cloud environments.

**Module IV:** Cloud Operations and DevOps- Automation in cloud, configuration management, container orchestration basics, CI/CD pipelines, cloud cost management, and introduction to serverless computing.

**Text Books:**

1. Cloud Computing: Concepts, Technology & Architecture Thomas Erl, Ricardo Puttini, and Zaigham Mahmood, *Cloud Computing: Concepts, Technology & Architecture*, Prentice Hall, 2013.
2. Cloud Computing Bible Barrie Sosinsky, *Cloud Computing Bible*, Wiley Publishing, 2011.

**Reference Books:**

1. Architecting the Cloud Michael J. Kavis, *Architecting the Cloud: Design Decisions for Cloud Computing Service Models (SaaS, PaaS, and IaaS)*, Wiley, 2014.

2. Cloud Native DevOps with Kubernetes John Arundel and Justin Domingus, *Cloud Native DevOps with Kubernetes*, O'Reilly Media, 2019.

Course Code	Course Name	L-T-P: C
IT362	Cloud Systems Management Lab	0-0-3: 2

**Objectives:** To provide hands-on experience in deploying, managing, and monitoring cloud infrastructure and services using modern cloud platforms and tools.

**Learning Outcomes:** Students will be able to:

1. Provision and manage cloud resources (VMs, storage, networking).
2. Deploy and monitor applications on cloud platforms.
3. Implement basic security and access control mechanisms.
4. Use containerization and automation tools for cloud operations.
5. Optimize and manage cloud resources efficiently.

**Tentative Lab Assignments:**

1. Introduction to a cloud platform (AWS/Azure/GCP) and basic account setup.
2. Creation and management of virtual machines (VMs).
3. Configuration of cloud storage (object and block storage).
4. Setting up virtual networks and basic network security (firewalls/security groups).
5. Deployment of a simple web application on a cloud instance.
6. Monitoring cloud resources using built-in monitoring tools.
7. Implementing Identity and Access Management (IAM) policies and roles.
8. Containerization of an application using Docker.
9. Basic container orchestration using Kubernetes (or managed services).
10. Automating deployment using scripts or CI/CD tools.
11. Exploring serverless computing (e.g., deploying a simple function).
12. Cost estimation and resource optimization in cloud environments.

## Semester-VII

### Courses and Contents

Course Code	Course Name	L-T-P: C
SI491	Research / Industrial Internship*	0-0-0: 3

\*The Research/Industrial Internship is a Pass/Fail course. It is conducted during summer vacation between 6<sup>th</sup> and 7<sup>th</sup> semester. The evaluation of the internship is carried out in 1<sup>st</sup> week of 7<sup>th</sup> semester and its grade will appear in the 7<sup>th</sup> semester grade sheet.

**Scope of the Internship:** It provides the students with an opportunity to update their skill-set by deploying themselves in a real-time environment. It is offered in two modes:

1. Summer Research Internship (SRI) and
2. Summer Industrial Internship (SII)

Under SRI mode, a student is allowed to join any academic institution around the globe by collaborating with faculty and associated research lab. It provides an opportunity to expose himself/herself with the initial glimpse of research environment in more profession way.

Under SII mode, a students is introduced to the corporate world. It teaches him/her professional ethics and polishes his/her soft skills like communication and interpersonal skills. This internship will be helpful for an effortless adaptation to work environment when he/she joins a full-time job. Also, a considerable number of internships come with a pre-placement offer (PPO) which gives an opportunity to prove his/her abilities and convert internship into a full-time job. In addition, SII allows to create professional network which could come in handy when a student will be applying for jobs in future.

Course Code	Course Name	L-T-P: C
PH401	Physics of Atoms and Molecules	3-0-0: 3

**Objectives:** The primary objective of this course is to provide a fundamental understanding of the quantum structure of atoms and molecules. Develop students' ability to relate microscopic phenomena to observable spectra.

**Learning Outcomes:** After successful completion of this course, students will be able to explain the quantum mechanical basis of atomic and molecular structure and

apply principles of atomic and molecular physics in spectroscopy, lasers, and material science.

**Prerequisites:** Basic knowledge of Quantum Mechanics, Classical Mechanics and Electromagnetism, Differential Equations, and Linear Algebra

#### Contents:

**Module-I:** Historical development of atomic models, Thomson, Rutherford, Bohr. Limitations of classical models and introduction to quantum ideas, Schrödinger equation for hydrogen atom, Quantum numbers and their physical meaning, Atomic orbitals and their shapes, Electron spin and Pauli's exclusion principle, Multi-electron atoms – Hartree and Hartree-Fock approximations (12 Lects)

**Module-II:** Origin of atomic spectra and selection rules, Spin-orbit coupling and fine structure, Zeeman effect, Paschen-Back effect, Stark effects, Hyperfine structure and isotope shifts, LS and jj coupling schemes, Alkali spectra and term symbols, X-ray spectra and Moseley's law (10 Lects)

**Module-III:** Introduction to molecular systems and Born–Oppenheimer approximation, Molecular orbitals vs. valence bond theory, Energy states, Molecular term symbols and electronic configurations, Hybridization and sigma–pi bonding, Potential energy curves and bond dissociation, Intermolecular forces (Van der Waals, dipole–dipole, hydrogen bonding), Electronic transitions in molecules (10 Lects)

**Module-IV:** Introduction to spectroscopy and types of spectra, Rotational spectra of diatomic molecules, Vibrational spectra (Infrared spectroscopy), Vibration–rotation spectra, Raman Effect and Raman spectroscopy (10 Lects)

#### Text Books:

1. B. H. Bransden and C. J. Joachain, Physics of Atoms and Molecules, Pearson Education (2003).
2. H. E. White, Introduction to Atomic Spectra, McGraw-Hill (1934).

#### Reference Books:

1. Condon, E. U. and Shortley, G. H., (1951), The Theory of Atomic spectra, Cambridge University Press.

## Semester-VIII

### Courses and Content

Course Code	Course Name	L-T-P: C
BP490	B. Tech. Project	0-0-36: 12

The B. Tech. Project (BTP) is an academic course. A student may pursue BTP

- On-campus:** Under the supervision of the Institute's faculty members.
- Off-campus:** At an Organization/Institute of repute under an Off-campus supervisor. In addition, student will be assigned On-campus mentor from the Institute. The Off-campus supervisor gives feedback to the On-campus mentor about the progress of the project and the On-campus mentor provides the interface to the student with the Institute.

#### Acceptable B. Tech. Projects:

- Software Development.
- System Design and Simulation.
- Hardware Development/Implementation.
- Embedded System (Software & Hardware combined) Development/Implementation.
- Theoretical Modeling.
- Technical Study.
- Modules of a big research and development project jointly guided by teams of faculty with a focus on synthesis of their class-room learning to solve real world problems.

#### The project work could be on

- A novel/new idea.
- An extension of some previous research work.
- An abstract problem.
- Some proof of concept problem.

**Objective:** The project work (On-campus or Off-campus)/Internship (Off-campus) is expected to provide an opportunity to final year undergraduate students to design/develop/implement systems involving information technology, computer science, embedded systems and its applications, requiring an individual effort on the part of the student.

**Scope:** Different kinds of projects and the associated deliverables that could be accepted as a final year student's project are conceived broadly as follows:

- Software Development.
- System Design & Simulation.
- Hardware Development/Implementation.

**Deliverables:** In general, the expected deliverables will include one or more of the followings

- Product developed.
- Code developed.
- Research work in the form of Conference/Journal paper.
- Software developed along with instructions and source code.
- Comparative study (products/ methods/ designs).
- Simulation study results.
- Project report at the end of the work.

In case certain projects may not get clearly classified amongst the categories mentioned above, the concerned supervisor may identify the specific deliverables for those projects.

**Group project:** A group project is defined as a project where several students work in a group on one problem; or sub-task of a larger problem or a problem set. In such case it is mandatory to clearly define the deliverables of individual student of the group.

**Duration:** The Off-campus project is for a minimum duration of 16 weeks after the completion of seventh semester. It is expected that the project would end by the last week of April in the respective academic year.

**Expectations from the student:** Followings are expected from students:

#### For On-campus students:

- Student must find an On-campus supervisor from IIIT Vadodara.
- The student is expected to complete the project work assigned by the On-campus supervisor and is expected to meet all the milestones identified.
- The student is expected to follow the work-plan decided by his/her On-campus supervisor. This includes reporting, leave and working hours during the project tenure.
- The student should report immediately to the On-campus supervisor in the event of exceptional circumstances like illness.
- The student is supposed to submit a 'Project Report' in the prescribed format to the BTP Coordinator at IIIT Vadodara on or before due date.
- Student will be responsible for the plagiarism and copy-right issues.

#### For Off-campus students:

- Student will be allotted an On-campus mentor from IIIT Vadodara. Students are advised to meet prospective faculty member to act as mentor.
- The student is expected to complete the project work assigned by the Off-campus supervisor and is expected to meet all the milestones identified.

3. The student is expected to follow the work-plan decided by his/her Off-campus supervisor and due concern with On-campus mentor. This includes reporting, leave and working hours during the project tenure.
4. The student should report immediately to the Off-campus supervisor and On-campus mentor in the event of exceptional circumstances like illness.
5. The student is expected to follow the rules and regulations of the organization as briefed by the Off-campus supervisor.
6. The student is supposed to submit a “Project Report” in the prescribed format to the BTP Coordinator at IIIT Vadodara on or before the due date.
7. Student will be responsible for the plagiarism and copy-right issues.

**Expectations from the Organization (Off-campus BTP):** Followings are expected from the Organization where the student is pursuing the project:

1. Provide an opportunity to the student to carry out a project that satisfies the objective, scope and guidelines of the final year student projects given above.
2. An Off-campus supervisor is to be assigned who would look after the project work of the student and interact closely with the student’s On-campus mentor at IIIT Vadodara.
3. The Off-campus supervisor is expected to supervise the performance of the student in achieving the required milestones and is advised to send the feedback on a regular basis (as decided mutually) to the On-campus mentor.
4. Student is required to defend/present his project work in the respective academic year. This requires the physical presence of the student on the campus and this presentation will formally close the process of BTP. The organization is expected to relieve the student by that time.
5. The Organization's Non-disclosure Agreement (NDA), if any, must not prohibit the student to show the data, technique and/or results to the evaluation committee during the presentation of the project.
6. Off-campus supervisor is required to provide evaluation details as listed in next section, to On-campus mentor on or before in the respective academic year.

**BTP Report:** The mid-semester BTP report should not exceed the prescribed length of 10 pages in the single column format with a font size 12 points and Times New Roman. The page limit of 10 pages will be strictly

imposed. Students are required to submit their mid semester report on or before 1<sup>st</sup> March of the respective academic year.

All care should be taken to write a final report that summarizes the work carried out by the student as part of his/her BTP. The report should not exceed the prescribed length of 40 pages in the single column format with a font size 12 points and Times New Roman. The page limit of 40 pages will be strictly imposed.

The BTP report submitted by the students will be passed through a *plagiarism check* using the Turnitin or similar anti-plagiarism software. Reports which bear similarity of more than 5% with a single source and a cumulative similarity of 20% will be identified and communicated to the evaluation committee and On-campus BTP supervisor/mentor. Note that subsequent revision of the BTP report is not possible under any circumstances.

## Appendix A

### Programme Electives (PE1-PE8) CSE/AI/IT/CSE (Cyber Security)

Course Code	Course Name	L-T-P: C
	Software and Cybersecurity	3-0-2: 4

**Objectives:** This course aims to provide a comprehensive understanding of software security and cybersecurity principles, focusing on identifying vulnerabilities, securing applications, and applying practical techniques such as penetration testing and web security to protect systems and data.

#### Learning Outcomes:

1. Understand fundamental concepts of cybersecurity and threat landscapes.
2. Identify common software vulnerabilities and security issues.
3. Apply penetration testing techniques to assess system security.
4. Analyze and mitigate web application security risks.
5. Use cybersecurity tools for vulnerability assessment and defense.

#### Contents:

**Module I:** Introduction: Fundamentals of cybersecurity: CIA triad (Confidentiality, Integrity, Availability)- Threats, vulnerabilities, and risk management, Types of cyberattacks (malware, phishing, DoS/DDoS, ransomware), Security policies, standards, and frameworks, Basics of cryptography and authentication mechanisms.

**Module II:** Software Security Issues- Secure software development lifecycle (SDLC), Common software vulnerabilities (buffer overflow, injection, XSS), OWASP Top 10 vulnerabilities, Input validation, error handling, and secure coding practices, Static and dynamic code analysis.

**Module III:** Penetration Testing- Introduction to penetration testing and ethical hacking, Phases: reconnaissance, scanning, exploitation, post-exploitation, Vulnerability assessment vs penetration testing, Basics of social engineering, Reporting and remediation strategies.

**Module IV:** Web Security, Web application architecture and attack surface Common web attacks: SQL injection, XSS, CSRF, session hijacking, Secure authentication and session management, HTTPS, SSL/TLS basics, Web security testing methodologies.

**Module V:** Cybersecurity Tools and Practices- Network scanning tools (e.g., Nmap), Vulnerability assessment tools (e.g., Nessus, OpenVAS), Web testing tools (e.g., Burp Suite), Basics of SIEM tools and log analysis, Incident response and digital forensics basics

#### Text Books:

1. Stallings, W. – Cryptography and Network Security: Principles and Practice, Pearson.
2. Goodrich, M.T. & Tamassia, R. – Introduction to Computer Security, Pearson.
3. Kaufman, C. et al. – Network Security: Private Communication in a Public World, Prentice Hall.

#### Reference Books:

1. Stamp, M. – Information Security: Principles and Practice, Wiley.
2. Erickson, J. – Hacking: The Art of Exploitation, No Starch Press.
3. Stuttard, D. & Pinto, M. – The Web Application Hacker’s Handbook, Wiley.
4. Anderson, R. – Security Engineering, Wiley.

Course Code	Course Name	L-T-P: C
	Advanced Multimedia Technologies	3-0-2: 4

**Objectives:** The objective of this course is to develop a comprehensive understanding of digital multimedia technologies, immersive environments, and their underlying computational models. It aims to equip students with the ability to analyze, design, and evaluate multimedia systems, including VR/AR and volumetric media. The course also focuses on visual quality assessment methods to ensure optimal user experience in emerging multimedia applications.

#### Learning Outcomes:

1. Analyze multimedia systems using both objective quality metrics and user-centric QoE measures.
2. Evaluate and model QoS parameters (latency, bandwidth, jitter) and their impact on QoE.
3. Design and optimize immersive systems considering perception, system constraints, and experience quality.
4. Apply advanced techniques for assessing and improving quality in VR/AR, 360°, and volumetric media.

#### Contents:

**Module I:** Digital and Interactive Multimedia Fundamentals of multimedia systems; depth estimation techniques; LiDAR sensing and RGB-D acquisition; lenslet imaging and light field representations; 360°

images and videos; projections and viewport mapping; voxelization and volumetric representations; point clouds and mesh-based models; basics of video compression and immersive codecs (H.264/HEVC/AV1, MPEG-I).

**Module II:** Virtual and Augmented Reality Systems Degrees of freedom (3DoF/6DoF); 3D transformations and rotations; rendering pipeline; depth of field and view-dependent rendering; gaze tracking (eye and head); interaction techniques; VR/AR hardware and software platforms; Unity-based 3D asset creation; distortions, latency, and limitations in XR systems.

**Module III:** Neural Rendering and 3D Scene Representation Introduction to neural rendering; Neural Radiance Fields (NeRF) for novel view synthesis; volumetric rendering principles; extensions of NeRF (fast NeRF, dynamic NeRF); 3D Gaussian Splatting for real-time rendering; comparison with traditional representations (meshes, point clouds, voxels); integration of neural methods with LiDAR and multi-view data; applications in immersive media and digital twins.

**Module IV:** Human Perception and Experience Modeling Human visual system (HVS); visual attention and saliency; motion perception; psychological and physiological aspects of immersion; cybersickness and latency effects; perceptual thresholds; user-centric design for VR/AR systems.

**Module V:** Visual Quality Assessment Subjective quality assessment (MOS/DMOS); objective metrics (Full Reference, Reduced Reference, No Reference); traditional metrics (PSNR, SSIM); perceptual and saliency-based quality models; quality assessment for immersive media (360°, point clouds, light fields, neural rendering outputs); dataset-based evaluation and benchmarking.

**Module VI:** Quality of Service (QoS) in Multimedia Systems Definition and role of QoS; key parameters: latency, bandwidth, jitter, packet loss; QoS in video streaming, XR, and cloud rendering; adaptive streaming (DASH, viewport-adaptive streaming); edge computing and 5G/6G support for immersive applications; system-level optimization.

**Module VII:** Quality of Experience (QoE) Modeling and Evaluation QoE concepts and frameworks; relationship between QoS and QoE; subjective vs objective QoE evaluation; user studies and perceptual experiments; QoE models for video, VR/AR, and gaming; ITU-T standards; machine learning-based QoE prediction; cross-layer optimization strategies.

**Module VIII:** Advanced Applications and Research Trends QoE/QoS in immersive media (360°, point clouds, NeRF-based rendering); cloud gaming and

metaverse systems; real-time rendering pipelines; digital twins and smart environments; case studies (Netflix, YouTube, XR platforms); emerging trends in neural rendering and immersive technologies.

**Tentative Laboratory Assignments:**

1. Implement depth estimation and point cloud visualization using open datasets.
2. Develop a basic VR/AR application in Unity incorporating gaze or interaction.
3. Perform subjective quality assessment (MOS) on a small multimedia dataset.
4. Implement and compare image/video quality metrics (PSNR, SSIM, etc.).
5. Evaluate quality for 360° images and point cloud datasets using appropriate metrics.

**Text Books:**

1. G. Petrovic, Point Cloud Processing and 3D Reconstruction. Cham, Switzerland: Springer, 2020.
2. J. Zhang, 3D Imaging, Analysis and Applications. Cham, Switzerland: Springer, 2020.
3. T. Dagiuklas, L. Atzori, C. W. Chen, and P. Chatzimisios, Multimedia Quality of Experience (QoE): Current Status and Future Requirements. Hoboken, NJ, USA: Wiley, 2017.
4. S. Winkler, Digital Video Quality: Vision Models and Metrics. Chichester, UK: Wiley, 2005.

**Reference Books:**

1. O. Schreer, P. Kauff, and T. Sikora, 3D Video Communication: Algorithms, Concepts and Real-Time Systems. Hoboken, NJ, USA: Wiley, 2005.
2. A. Smolic et al., “Light Field Imaging and Display: State-of-the-Art,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 27, no. 12, pp. 2556–2572, 2017.
3. M. Preda and F. Dufaux, “Point Cloud Compression and Quality Assessment,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 29, no. 10, pp. 2899–2913, 2019.
4. ITU-T, Recommendations on Multimedia Quality, QoE, and Subjective Assessment (P.910, P.1203, G.100).

Course Code	Course Name	L-T-P: C
	Recommender Systems	3-0-2: 4

**Objectives:** This course explores the core concepts and methodologies of recommender systems (collaborative, content-based, hybrid). It focuses on designing,

implementing, and assessing algorithms using authentic datasets and developing strategies to overcome challenges like data sparsity and the cold-start problem. Advanced topics include context-aware and deep learning-based systems.

**Learning Outcomes:**

1. Understand and implement core recommender system techniques such as collaborative and content-based filtering.
2. Analyze and evaluate recommender systems using appropriate metrics and datasets.
3. Design and develop recommendation models for real-world applications.
4. Apply advanced methods including matrix factorization, deep learning, and context-aware recommendations.

**Contents:**

**Module I: Introduction to Recommender Systems-** Overview of recommender systems, applications in e-commerce, streaming, and social media; types of recommendation systems; challenges such as cold start, sparsity, and scalability; data collection (explicit vs implicit feedback) and preprocessing techniques.

**Module II: Collaborative and Content-Based Filtering-** User-based and item-based collaborative filtering; similarity measures (cosine, Pearson correlation); content-based recommendation techniques; feature extraction and user profiling; advantages and limitations of each approach.

**Module III: Matrix Factorization and Hybrid Methods-** Latent factor models; singular value decomposition (SVD), SVD++; alternating least squares (ALS); hybrid recommender systems combining collaborative and content-based methods; addressing sparsity and cold-start problems.

**Module IV: Evaluation and Advanced Techniques-** Evaluation metrics (RMSE, MAE, precision, recall, F1-score, NDCG); offline and online evaluation; context-aware recommender systems; trust-based and social recommendation; introduction to deep learning-based recommender systems (neural collaborative filtering, autoencoders).

**Module V: Scalability and Real-World Systems-** Large-scale recommendation systems; distributed computing frameworks; real-time recommendations; recommendation pipelines; case studies from industry (Netflix, Amazon, YouTube).

**Tentative Laboratory Assignments:**

1. Implement user-based and item-based collaborative filtering using a real dataset (e.g., MovieLens).

2. Develop a content-based recommender using feature extraction techniques.
3. Implement matrix factorization (SVD/ALS) for recommendation tasks.
4. Evaluate recommender systems using metrics such as RMSE, precision, and recall.
5. Build a hybrid recommender system and compare its performance with individual models.

**Text Books:**

1. Francesco Ricci, Lior Rokach, Bracha Shapira, Bracha Shapira, Springer 2011 edition.
2. Dietmar Jannach, Markus Zanker, Alexander Felfernig, Recommender Systems: An Introduction, Cambridge University Press 1st edition.

**Reference Books:**

1. M.D. Ekstrand, J.T. Riedl, J.A. Konstan, Collaborative filtering recommender systems.
2. X. Su, T.M. Khoshgoftaar, A survey of collaborative filtering techniques, Adv. Artif. Intell., 2009.
3. Y. Koren, Factorization meets the neighbourhood: a multifaceted collaborative filtering model, in: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2008, pp. 426–434.

Course Code	Course Name	L-T-P: C
	Visual Quality Assessment	3-0-2: 4

**Objectives:** The objective of this course is to provide a comprehensive understanding of visual saliency and quality assessment techniques for images, videos, and immersive media. It aims to equip students with the ability to design, evaluate, and analyze objective and subjective quality assessment models. The course also introduces emerging multimedia formats and standards, enabling research-oriented exploration in modern visual computing.

**Learning Outcomes:**

1. Analyze and implement visual saliency estimation methods using classical and learning-based approaches.
2. Evaluate quality assessment models using appropriate metrics, datasets, and subjective testing methodologies.
3. Design and compare full-reference, reduced-reference, and no-reference quality assessment techniques.

4. Apply quality assessment methods to emerging media formats such as 360° images, light fields, and point clouds.

#### Contents:

**Module I:** Saliency Estimation: This module will introduce the fundamental concepts of visual saliency and quality assessment. We would start with visual saliency estimation and its challenges. Broad range of applications where estimating saliency would be essential. The different approaches for saliency estimation would be discussed. The traditional top-down and bottom-up approaches would be emphasized. Apart from this, the object-based, content-based, context-based, and task-based approaches for saliency estimation would also be introduced.

**Module II:** Evaluation Strategies: The different evaluation metrics used to validate a saliency estimation model would be identified. Importance of the different evaluation metrics would be studied. We would be learning how a visual quality dataset is created and would also explore a publicly available dataset for evaluating the saliency models. Strategies and recommendations on subjective tests would be learned.

**Module III:** Quality Estimation- What is Image quality estimation and why it is important. Broad domains where quality estimation is essential. Introduction to full-reference, reduced-reference, and no-reference quality estimation models. Traditional approaches such as PSNR and SSIM. Importance of saliency in quality estimation. Deep learning-based approaches in quality estimation.

**Module IV:** Evaluation Strategies- Metrics used to evaluate a quality estimation model. Explore a publicly available dataset and evaluate certain quality estimation models. ITU-T recommendations on quality assessment of visual information.

**Module V:** Introduction to New Multimedia Technologies- Omnidirectional images and videos, Light field images, LIDAR images, Point cloud data.

#### Tentative Laboratory Assignments:

1. Implement and compare saliency estimation methods (e.g., Itti-Koch model vs. deep learning-based models) on benchmark datasets.
2. Evaluate saliency models using standard metrics (AUC, NSS, KL-divergence) on publicly available datasets.
3. Implement image quality assessment techniques (PSNR, SSIM) and compare with deep learning-based approaches.

4. Perform quality assessment on immersive media (e.g., 360° images or point clouds) using standard datasets and evaluation protocols.

#### Text Books:

1. A. C. Bovik, *Handbook of Image and Video Processing*, 2nd ed. Burlington, MA, USA: Academic Press, 2005.
2. Z. Wang and A. C. Bovik, *Modern Image Quality Assessment*. San Rafael, CA, USA: Morgan & Claypool, 2006.

#### Reference Books:

1. ITU-T, *Recommendations on Subjective and Objective Quality Assessment of Multimedia*, various standards (e.g., P.910, P.1203).
2. A. Borji and L. Itti, "State-of-the-Art in Visual Attention Modeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 185–207, 2013.
3. M. Narwaria, M. Perreira Da Silva, and P. Le Callet, "HDR-VQM: An Objective Quality Measure for High Dynamic Range Video," *Signal Processing: Image Communication*, 2015.
4. Datasets: LIVE, TID2013, SALICON, and other publicly available benchmarks for saliency and quality assessment.

Course Code	Course Name	L-T-P: C
	Introduction to Metaverse	3-0-2: 4

**Objectives:** The course involves exploring the technologies, concepts, and applications that underpin the rapidly evolving digital landscape.

**Learning Outcomes:** By the end of the course, participants will:

1. Understand the core concepts and technologies driving the Metaverse.
2. Gain hands-on experience with tools for creating virtual worlds.
3. Learn about the societal, economic, and ethical implications of the Metaverse.
4. Develop skills to conceptualize and build Metaverse applications.

#### Contents:

**Module I: Introduction to the Metaverse-** What is the Metaverse? Origins and evolution, Key components: Virtual reality (VR), augmented reality (AR), mixed reality (MR), and Web3, Difference between the Metaverse and traditional virtual worlds, Major players in the Metaverse (Meta, Decentraland, Roblox, etc.),

Case studies of Metaverse applications, Discussion: Potential and limitations of the Metaverse.

**Module II: Foundations of Virtual and Augmented Reality-** Overview of VR and AR technologies, Hardware and software essentials (VR headsets, AR glasses, and XR platforms), Introduction to Unity3D and Unreal Engine for VR/AR development, Handson: Creating a basic VR/AR experience using Unity or Unreal Engine.

**Module III: Building Virtual Worlds-** Basics of 3D modeling for the Metaverse using Blender, Environment design and user interaction principles, Optimizing virtual spaces for performance and scalability, Handson: Designing a virtual environment using Unity3D or Mozilla Hubs.

**Module IV: Avatars and Digital Identity-** Creating avatars and digital representations, Identity management and personalization in the Metaverse, Interoperability of avatars across platforms, Handson: Customizing avatars using Ready Player Me or VRoid Studio.

**Module V: Social Interaction and Community Building-** Role of social networks in the Metaverse, Collaborative tools and virtual meetings (e.g., Spatial, Gather.Town, Horizon Workrooms), Challenges of virtual societies (e.g., moderation and governance), Simulation: Hosting a virtual event in Mozilla Hubs or AltspaceVR.

**Module VI: Monetization and the Metaverse Economy-** Blockchain and Web3 in the Metaverse, Role of blockchain in the Metaverse: Decentralization and interoperability, Cryptocurrencies and NFTs as digital assets, Smart contracts and Decentralized Autonomous Organizations (DAOs), Handson: Creating and deploying a simple NFT using platforms like OpenSea or Rarible, Revenue streams: Virtual goods, real estate, events, and services, Introduction to playtolearn gaming models, Marketplaces for digital assets (NFT marketplaces, virtual land), Workshop: Building a virtual shop or service in a Metaverse platform.

**Module VII: Ethics, Privacy, and Security-** Ethical considerations: Accessibility, equity, and addiction, Privacy concerns in immersive environments, Securing digital assets and user data, Case study: Analyzing ethical dilemmas in Metaverse projects.

**Module VIII: Future Trends and Applications-** AI and machine learning in the Metaverse, Role of 5G and edge computing, Emerging use cases: Healthcare, education, entertainment, and beyond, Group project: Pitching a Metaversebased startup idea.

#### Tentative Laboratory Assignment:

1. **VR/AR Environment Development:** Create a basic immersive VR or AR application using Unity3D or Unreal Engine, including simple navigation and object interaction.
2. **3D Virtual World Design:** Design and deploy a virtual environment using Blender and integrate it into a platform such as Unity3D or Mozilla Hubs, focusing on usability and performance.
3. **Avatar Creation and Interaction:** Develop and customize a personal avatar using tools like Ready Player Me or VRoid Studio, and implement basic interactions within a virtual space.
4. **Social Interaction Simulation:** Host and manage a virtual meeting or event using platforms like Mozilla Hubs or Gather.Town, incorporating collaboration features and user engagement elements.
5. **NFT and Metaverse Economy Implementation:** Create a simple digital asset (NFT), deploy it on a marketplace (e.g., OpenSea), and design a basic virtual storefront or service within a metaverse platform.

#### Text Books:

1. M. Ball, *The Metaverse: And How It Will Revolutionize Everything*. New York, NY, USA: Liveright Publishing, 2022.
2. T. Parisi, *Learning Virtual Reality: Developing Immersive Experiences and Applications for Desktop, Web, and Mobile*. Sebastopol, CA, USA: O'Reilly Media, 2015.

#### Reference Books:

1. D. Schmalstieg and T. Höllerer, *Augmented Reality: Principles and Practice*. Boston, MA, USA: Addison-Wesley, 2016.
2. ACM SIGGRAPH, "Conference Proceedings and Courses," ACM, various years. [Online]. Available: <https://www.siggraph.org>

Course Code	Course Name	L-T-P: C
	<b>Generative Artificial Intelligence and Large Language Models</b>	<b>3-0-2: 4</b>

**Objectives:** This course will be providing a comprehensive understanding of Generative AI and Large Language Models (LLMs), focusing on their theoretical foundations, architectures, training paradigms, and real-world applications. The course will equip students with practical skills in building, fine-tuning, and deploying generative models for text,

image, and multimodal applications, while addressing ethical, fairness, and societal implications.

#### Learning Outcomes:

1. Understand the fundamentals of Generative AI and probabilistic modeling.
2. Explain architectures of modern LLMs such as transformers and attention mechanisms.
3. Implement and fine-tune LLMs for downstream tasks using modern frameworks.
4. Design prompt engineering strategies and build LLM-powered applications.
5. Analyze ethical, bias, and safety concerns in generative AI systems.
6. Develop end-to-end generative AI solutions using real-world datasets and tools.

#### Contents:

##### Module I: Introduction to Generative AI

1. Fundamentals of Generative AI vs Discriminative models.
2. Probability distributions, latent variables, and generative modeling.
3. Overview of generative models: Autoencoders, Variational Autoencoders (VAEs), GANs.
4. Introduction to Large Language Models (LLMs).
5. Applications of Generative AI (text, image, code, healthcare, design)

##### Module II: Transformers and Large Language Models

1. Limitations of RNNs and LSTMs.
2. Transformer architecture: self-attention, multi-head attention, positional encoding.
3. Encoder-decoder vs decoder-only models.
4. Pretrained LLMs: GPT, BERT, T5 (conceptual understanding).
5. Scaling laws and emergent capabilities

##### Module III: Training and Fine-Tuning LLMs

1. Pretraining objectives: language modeling, masked language modeling.
2. Fine-tuning techniques: full fine-tuning vs parameter-efficient tuning (LoRA, adapters).
3. Instruction tuning and alignment.
4. Reinforcement Learning from Human Feedback (RLHF).
5. Dataset preparation and tokenization

##### Module IV: Prompt Engineering and LLM Applications

1. Prompt design: zero-shot, few-shot, chain-of-thought prompting.
2. Retrieval-Augmented Generation (RAG).
3. Embeddings and semantic search.

4. Building applications using LLM APIs (chatbots, summarization, QA systems).
5. Tools and frameworks: Hugging Face, LangChain, vector databases

##### Module V: Generative AI for Multimodal Systems and Ethics

1. Image generation models: GANs, Diffusion models.
2. Multimodal AI: text-to-image, vision-language models.
3. Evaluation metrics for generative models.
4. Bias, fairness, hallucination in LLMs.
5. Privacy, security, and responsible AI.
6. Legal and societal implications of Generative AI

#### Practical (Lab Component)

1. **Implement basic text generation using pretrained LLMs:** Use Hugging Face/Open-source models for text generation.
2. **Tokenization and Embedding Analysis:** Explore tokenizers (BPE/WordPiece), Generate and visualize embeddings.
3. **Prompt Engineering Techniques:** Zero-shot, few-shot, chain-of-thought prompting, Performance comparison.
4. **Fine-tune a transformer model on custom dataset:** Supervised fine-tuning using domain-specific data.
5. **Parameter-Efficient Fine-Tuning (PEFT):** Implement LoRA/adapters, Compare with full fine-tuning.
6. **Build a chatbot using LLM APIs:** Add memory and contextual conversation.
7. **Implement RAG-based document QA system:** Use embeddings + vector database (FAISS).
8. **Evaluation of Generated Text:** Metrics: BLEU, ROUGE, Human vs automated evaluation.
9. **Image generation using diffusion models:** Text-to-image generation using Stable Diffusion.
10. **Mini Project: End-to-end Generative AI application:** Examples: chatbot, content generator, domain-specific assistant.

#### Text Books:

1. Deep Learning – Ian Goodfellow, Yoshua Bengio, Aaron Courville.
2. Natural Language Processing with Transformers – Lewis Tunstall et al.
3. Speech and Language Processing – Daniel Jurafsky & James H. Martin

#### Reference Books:

1. Generative Deep Learning – David Foster.
2. Pattern Recognition and Machine Learning – Christopher Bishop.

Course Code	Course Name	L-T-P: C
	<b>Data Science in Healthcare</b>	<b>3-0-2: 4</b>

**Objective:** This course aims to provide a comprehensive understanding of data science techniques applied to healthcare and biomedical domains. It focuses on medical data types, predictive modeling, clinical decision support systems, and AI-driven healthcare analytics. The course emphasizes real-world applications such as disease diagnosis, survival prediction, medical imaging analysis, and precision medicine, along with ethical and regulatory considerations.

**Learning Outcomes:**

1. Understand different types of healthcare data (clinical, imaging, genomic, wearable).
2. Apply machine learning and statistical methods for healthcare analytics.
3. Develop predictive models for diagnosis, prognosis, and survival analysis.
4. Analyze medical imaging data using AI techniques.
5. Interpret models using explainable AI methods in clinical settings.
6. Understand ethical, legal, and regulatory aspects of healthcare data science.

**Contents:**

**Module I: Introduction to Healthcare Data Science**

1. Overview of healthcare systems and digital health.
2. Types of healthcare data: EHR/EMR, imaging, genomics, sensor data.
3. Data characteristics: heterogeneity, missing data, imbalance.
4. Introduction to clinical workflows and decision-making.
5. Case studies in AI-driven healthcare

**Module II: Data Preprocessing and Statistical Analysis**

1. Data cleaning, normalization, and feature engineering.
2. Handling missing and imbalanced data.
3. Exploratory data analysis (EDA) in healthcare.
4. Statistical methods: hypothesis testing, regression, survival basics.
5. Evaluation metrics: sensitivity, specificity, ROC-AUC.

**Module III: Machine Learning for Healthcare**

1. Supervised learning: classification (SVM, Logistic Regression, Random Forest).

2. Unsupervised learning: clustering, anomaly detection.
3. Feature selection and dimensionality reduction.
4. Predictive modeling for disease diagnosis and prognosis.
5. Case study: cancer survival prediction

**Module IV: Deep Learning and Medical Imaging**

1. Introduction to deep learning in healthcare.
2. CNN architectures for medical image analysis.
3. Segmentation, classification, detection in medical imaging.
4. Multimodal learning: combining imaging + clinical data.
5. Case study: brain tumor (Glioblastoma) analysis and radiomics

**Module V: Advanced Topics and Ethical Considerations**

1. Explainable AI (XAI): SHAP, LIME.
2. Bias, fairness, and generalization in healthcare AI.
3. Federated learning and privacy-preserving AI.
4. Clinical validation and translational AI.
5. Regulatory frameworks (HIPAA, GDPR – conceptual overview).
6. Future trends: precision medicine, digital twins

**Practicals:**

1. **Preprocessing real healthcare datasets (EHR/imaging):** Data cleaning, normalization, handling missing values.
2. **Exploratory Data Analysis (EDA) for Healthcare Data:** Statistical summaries, visualization, correlation analysis, Clinical interpretation of patterns.
3. **Feature Engineering and Selection in Medical Data:** Feature extraction, dimensionality reduction (PCA), Selection techniques (filter, wrapper methods).
4. **Implementation of classification models for disease prediction:** Logistic Regression, SVM, Random Forest.
5. **Handling Imbalanced Medical Data:** SMOTE, undersampling, class weighting, Evaluation with appropriate metrics.
6. **Survival analysis using clinical datasets:** Kaplan-Meier curves, Cox regression.
7. **Medical image analysis using CNNs (segmentation/classification):** Basic deep learning pipeline for imaging.
8. **Multimodal Learning (Clinical + Imaging Data Integration):** Combine structured and imaging data, Build hybrid predictive models.
9. **Explainability analysis using SHAP/LIME:** Model interpretation in clinical context.
10. **Mini Project: End-to-end healthcare AI solution**

**Text Books:**

1. Introduction to Statistical Learning – Gareth James et al.
2. Deep Learning – Ian Goodfellow et al.
3. Biomedical Informatics – Edward H. Shortliffe & James J. Cimino

**Reference Books:**

1. Pattern Recognition and Machine Learning – Christopher Bishop.
2. Hands-On Machine Learning with Scikit-Learn Keras and TensorFlow – Aurélien Géron.

Course Code	Course Name	L-T-P: C
	Agentic AI and Autonomous Systems	3-0-2: 4

**Objectives:** This course will provide a comprehensive understanding of Agentic AI and Autonomous Systems, focusing on their theoretical foundations, agent architectures, reasoning paradigms, and real-world applications. The course will equip students with practical skills in building, orchestrating, and deploying AI agents using modern frameworks for task automation, multi-agent collaboration, and vision-language applications, while addressing ethical, safety, and societal implications of autonomous AI systems.

**Learning Outcomes:**

1. Understand the fundamentals of Agentic AI and the sense-plan-act loop in autonomous systems.
2. Explain different agent architectures including ReAct, Chain-of-Thought, and memory-based reasoning pipelines.
3. Implement tool-using agents with capabilities such as web search, code execution, and external API integration.
4. Design and orchestrate multi-agent systems for automating complex real-world workflows.
5. Develop vision-language agents using multimodal models for image-guided understanding and decision making.
6. Apply safety, evaluation, and responsible deployment practices to agentic AI systems.

**Contents:**

**Module I- Foundations of AI & Large Language Models:** Introduction to Artificial Intelligence and Machine Learning, types of AI systems (narrow vs general), overview of Deep Learning, what are Large Language Models (LLMs), how LLMs are trained (pre-training and fine-tuning), tokenization, context window, temperature and sampling, how ChatGPT /

Gemini works, prompt engineering (zero-shot, few-shot, chain-of-thought), structured output from LLMs, hands-on: first API call in Python

**Module II- AI Agents:** Concepts & Reasoning Agent vs chatbot distinction, types of agents, PEAS framework, sense-plan-act loop, ReAct framework, Chain-of-Thought (CoT), Tree-of-Thought, planning agents, Reflexion, scratchpad reasoning, memory types (working, episodic, semantic), function calling and tool use basics

**Module III- Agent Frameworks & Building Agents:** LangChain agents, LlamaIndex, AutoGen, CrewAI, tool integration (web search, code execution, APIs), web browsing agents, RAG-based agents, building complete single agents with real tools, hands-on: end-to-end agent pipeline

**Module IV- Multi-Agent Systems:** Agent communication protocols, orchestration vs sub-agents, role-based agents, collaborative agents, conflict resolution, agent trust and safety, hands-on: multi-agent workflow using CrewAI / AutoGen

**Module V- Vision Agents & Multimodal AI:** Vision-language models (GPT-4V, LLaVA, Gemini), visual question answering, image-guided agents, multimodal RAG, multimodal pipelines, practical use cases: document scanner, product inspector, image captioning agent, hands-on: vision agent demo

**Module VI- Deployment, Ethics & Real-World Projects:** Agent evaluation metrics, guardrails, prompt injection attacks, agent memory management, productionizing agents, responsible AI and hallucinations, limitations of agents, case studies: coding agent, research agent, document agent, mini project: personal research assistant

**List of Experience:**

1. **Setting up the Environment-** Install Python, set up virtual environment, install LangChain, OpenAI SDK, and required libraries, configure API keys, verify setup with a hello world LLM call.
2. **First Interaction with an LLM-** Write a Python program to send prompts to an LLM using OpenAI / Gemini API, explore parameters like temperature, max tokens, and system prompt, observe how output changes with different settings.
3. **Prompt Engineering Techniques-** Implement and compare zero-shot, few-shot, and chain-of-thought prompting on the same problem, observe difference in output quality, document findings.
4. **Tool Integration with LLM-** Give an LLM access to external tools, a calculator, a weather API, and a web search tool, implement function calling, observe how the model decides when to use which tool.

5. **Building a Memory-based Chatbot-** Build a conversational chatbot using LangChain that remembers previous messages, implement short-term memory using conversation buffer, test multi-turn dialogue.
6. **RAG-based Document Q&A Agent-** Load a PDF document, convert to vector embeddings, store in a vector database (ChromaDB / FAISS), build an agent that answers questions strictly from the document.
7. **Multi-Agent System using CrewAI-** Define three agents, Researcher, Writer, and Reviewer, assign tasks and roles, run a collaborative pipeline where each agent contributes to a final output.
8. **Code Generation and Review Agent-** Build an agent that takes a problem statement, writes Python code, executes it, checks for errors, and iterates until correct output is produced.
9. **Vision Agent using Multimodal LLM-** Upload an image to GPT-4V or Gemini Vision, build an agent that answers questions about the image, extend to a document scanner that extracts text and tables from an image.
10. **End-to-End Mini Project-** Students implement one of the assigned mini projects integrating concepts from all previous experiments- RAG, tool use, memory, multi-agent or vision capabilities, followed by demonstration and viva

**Text Books:**

1. Artificial Intelligence: A Modern Approach – Russell & Norvig, Pearson, 2022 (4th Ed).
2. Building LLM Powered Applications – Valentina Alto, Packt, 2024.
3. Prompt Engineering for LLMs – John Berryman & Albert Webber, O'Reilly, 2024.
4. AI Agents in Action – Micheal Lanham, Manning, 2024.

**Reference Books:**

1. Designing Machine Learning Systems – Chip Huyen, O'Reilly, 2022.
2. LangChain, CrewAI & OpenAI Official Documentation, Online, 2024–25.

Course Code	Course Name	L-T-P: C
	<b>Computer Vision and Image Processing</b>	<b>3-0-0: 3</b>

**Objectives:** The objective of this course is to develop a comprehensive understanding of computer vision and image processing techniques, and their underlying computational and mathematical models. It aims to equip students with the ability to analyze, design, and

implement vision systems for real-world applications. The course focuses on both classical image processing methods and modern deep learning-based approaches, enabling students to build end-to-end pipelines for image classification, object detection, segmentation, and 3D vision tasks.

**Learning Outcomes:**

1. Analyze and apply fundamental image processing operations including filtering, edge detection, and morphological transformations.
2. Extract and match visual features using classical descriptors such as SIFT, ORB, and HOG for image correspondence tasks.
3. Design and train deep convolutional neural networks for image classification, object detection, and segmentation.
4. Implement 3D vision techniques including stereo matching, depth estimation, and point cloud processing.
5. Evaluate and deploy computer vision solutions for real-world applications in healthcare, surveillance, and autonomous systems.

**Contents:**

**Module I:** Introduction to Computer Vision: Human vs machine vision; image formation and camera models; pinhole camera model; lens distortion; image digitization and sampling; color spaces (RGB, HSV, YCbCr, Lab); pixel-level operations; image histograms; brightness and contrast manipulation.

**Module II:** Image Processing Fundamentals: Spatial domain filtering (Gaussian, Median, Bilateral); frequency domain processing and Fourier transforms; edge detection operators (Sobel, Laplacian, Canny); image thresholding (global, adaptive, Otsu); morphological operations (erosion, dilation, opening, closing); histogram equalization and CLAHE.

**Module III:** Feature Extraction and Matching: Corner detection (Harris, Shi-Tomasi); blob detection; scale-invariant feature transform (SIFT); SURF and ORB descriptors; Histogram of Oriented Gradients (HOG); feature matching strategies; RANSAC algorithm; homography estimation; image stitching and panorama creation.

**Module IV:** Deep Learning for Computer Vision: Introduction to Convolutional Neural Networks (CNNs); CNN architectures (LeNet, AlexNet, VGG, ResNet, EfficientNet); transfer learning and fine-tuning; image classification pipelines; object detection frameworks (YOLO, SSD, Faster R-CNN); anchor boxes and non-maximum suppression; performance evaluation metrics (mAP, IoU).

**Module V: Image Segmentation and 3D Vision:** Semantic segmentation (FCN, UNet, DeepLab); instance segmentation (Mask R-CNN); panoptic segmentation; stereo vision and epipolar geometry; disparity maps and depth estimation; structure from motion (SfM); point clouds and 3D reconstruction; depth sensors and RGB-D imaging.

**Module VI: Advanced Topics and Applications:** Optical flow estimation (Lucas-Kanade, Farneback); video understanding and action recognition; face detection and recognition (ArcFace, FaceNet); generative models in vision (GANs, Diffusion for image synthesis); medical image analysis; vision for autonomous driving; ethical considerations in vision systems.

#### Tentative Laboratory Assignments:

1. Implement basic image operations - color space conversion, histogram equalization, and pixel-level transformations using OpenCV.
2. Apply spatial filters and edge detection operators (Gaussian, Canny, Sobel) on real images and compare results.
3. Implement feature extraction using SIFT/ORB and perform feature matching between two images.
4. Train a CNN for image classification on image dataset using PyTorch / TensorFlow.
5. Implement real-time object detection using YOLOv8 on a custom or public dataset.
6. Perform semantic segmentation using UNet on a medical or satellite image dataset.
7. Implement stereo vision pipeline to compute disparity map and estimate depth from a stereo image pair.
8. End-to-End Mini Project - Real-time vision application (e.g., face mask detector, lane detection system, or plant disease classifier).

#### Text Books:

1. Richard Szeliski, Computer Vision: Algorithms and Applications, 2nd Ed. Springer, 2022.
2. Mohamed Elgendy, Deep Learning for Vision Systems. Manning, 2020.
3. Rafael C. Gonzalez and Richard E. Woods, Digital Image Processing, 4th Ed. Pearson, 2018.

#### Reference Books:

1. Gary Bradski and Adrian Kaehler, Learning OpenCV 4. O'Reilly, 2019.
2. Richard Hartley and Andrew Zisserman, Multiple View Geometry in Computer Vision, 2nd Ed. Cambridge University Press, 2004.
3. Jan Erik Solem, Programming Computer Vision with Python. O'Reilly, 2012.

Course Code	Course Name	L T P C
	<b>Time Series Analysis and Forecasting using Machine Learning and Deep Learning</b>	<b>3 0 0 3</b>

**Objectives:** To provide students with a comprehensive understanding of time series data analysis and forecasting techniques, covering fundamental concepts, statistical and machine learning models, evaluation methodologies, and deep learning approaches, enabling them to design and implement effective forecasting solutions for real-world temporal data applications.

**Learning Outcomes:** Upon successful completion of this course, students will be able to:

1. Understand the fundamentals of time series data, including its components, characteristics, and differences from traditional machine learning data.
2. Apply preprocessing techniques such as decomposition, detrending, differencing, and transformation to prepare time series data for modeling.
3. Analyze and evaluate forecasting models using appropriate validation strategies, baseline methods, and performance metrics.
4. Develop and implement classical statistical models such as autoregressive, moving average, ARIMA, and seasonal ARIMA for time series forecasting.
5. Design and build machine learning and deep learning models, including regression, MLP, CNN, and LSTM architectures, for solving real-world time series forecasting problems.

**Learning Outcomes:** After completion of the course, students will be able to:

1. Understand fundamental concepts of time series data, components, and forecasting challenges.
2. Apply preprocessing techniques such as decomposition, detrending, and stationarity transformation.
3. Analyze and evaluate forecasting models using validation strategies and performance metrics.
4. Develop classical statistical models such as AR, MA, ARIMA, and SARIMA.
5. Design and implement machine learning and deep learning models for real-world time series forecasting problems.

**Pre Requisites:** Machine Learning

#### Contents:

**Module 1: Fundamentals of Time Series-** Introduction to time series data including definitions, terminology, and real-world applications, understanding the components of time series such as trend,

seasonality, cyclicity, and noise, visualization and exploratory analysis of temporal data, differences between time series data and traditional machine learning datasets, transformation of time series into supervised learning problems using sliding window techniques, handling univariate and multivariate time series, and approaches for single-step and multi-step forecasting.

**Module 2: Temporal Structure and Preprocessing-** Concepts of white noise and random walk processes, understanding stationarity and its importance in time series modeling, decomposition of time series into additive and multiplicative components, techniques for removing trends including differencing and model-based approaches, handling seasonality through seasonal adjustment methods, data transformations such as scaling and logarithmic transformations to stabilize variance and improve model performance.

**Module 3: Forecast Evaluation and Problem Framing-** Different forecasting strategies including one-step and multi-step forecasting as well as direct and recursive approaches, backtesting methodologies such as train-test split and rolling window validation, evaluation metrics for forecasting performance including MAE, RMSE, MAPE, and related measures, development of baseline models such as naive, seasonal naive, and persistence models, analysis of residual errors and diagnostic checking, and reframing forecasting problems into regression and classification settings for machine learning applications.

**Module 4: Classical Forecasting Models-** Autoregressive and moving average modeling approaches for time series forecasting, integration of models into ARMA and ARIMA frameworks, extension to seasonal models such as SARIMA, application of the Box-Jenkins methodology including model identification, parameter estimation, and diagnostic validation, use of autocorrelation and partial autocorrelation for model selection, hyperparameter tuning techniques including grid search, and estimation of forecast intervals along with uncertainty analysis.

**Module 5: Machine Learning and Deep Learning for Time Series-** Framing time series forecasting problems for machine learning models, feature engineering techniques for temporal data, application of machine learning algorithms such as linear regression, decision trees, and ensemble methods, deep learning approaches including multilayer perceptrons, convolutional neural networks, and recurrent neural networks such as LSTM, model optimization and regularization strategies, handling multivariate time series data, and exploration of real-world applications including energy forecasting, financial time series prediction, and healthcare signal analysis.

#### Textbook:

1. Forecasting: Principles and Practice, Hyndman, R. J., & Athanasopoulos, G., *Forecasting: Principles and Practice* (3rd Edition), OTexts.

#### Reference Books:

1. Forecasting Methods and Applications, Makridakis, S., Wheelwright, S. C., & Hyndman, R. J., *Forecasting Methods and Applications* (3rd Edition), Wiley.
2. Time Series Analysis and Its Applications, Shumway, R. H., & Stoffer, D. S., *Time Series Analysis and Its Applications* (4th Edition), Springer.
3. Deep Learning for Time Series Forecasting, Brownlee, J., *Deep Learning for Time Series Forecasting*, Machine Learning Mastery.

Course Code	Course Name	L T P C
	<b>Block chain Technology and Applications</b>	<b>3 0 0 3</b>

**Objectives:** To equip students with comprehensive knowledge of block chain technology, from fundamentals and consensus mechanisms to architectural integration, quality evaluation, and real-world case studies

**Learning Outcomes:** Upon successful completion of this course, students will be able to:

1. Understand blockchain fundamentals, including applications, platforms like Bitcoin and Ethereum, and varieties such as decentralization and ledger structures.
2. Study and compare consensus algorithms like PoS, Algorand, and Ouroboros, and grasp smart contract basics, lifecycle, challenges, and use cases like supply chains.
3. Analyze blockchain's role in software architecture, covering elements like storage, communication, patterns, and integration as system components.
4. Evaluate blockchain's quality impacts, including cost models, performance, latency prediction, and properties like confidentiality, integrity, and reliability.
5. Study and examine real-world case studies, such as AgriDigital for agricultural supply chains and SecureVote prototype with token voting implementation.

**Prerequisites:** Basic Knowledge of Cryptography

**Contents:**

**Module I:** Introduction-Blockchain, Applications, Functionalities, Non Functional Properties, Architectural Design, Blockchain Platforms- Bitcoin, Ethereum, Hyperledger Fabric, Other Representative Block Chain Platforms, Varieties of Blockchain-Fundamental Properties, Decentralisation, Ledger Structure

**Module II:** Consensus Algorithm-Study and comparison of different consensus algorithms: PoS, PoS, Algorand, Ouroboros, Ethereum’s consensus, Block Configuration, Auxiliary Block Chains, Smart Contract Fundamentals: Introduction to Smart Contracts, Framework of smart contract, Life cycle of smart contract, Challenges of Smart Contract, Example Use Cases-Supply Chains.

**Module III:** Blockchain in Software Architecture-Architectural, storage, Computational Element, Communication Mechanism, Asset Management and Control Mechanism, Integrating Blockchain into a system as a component, Blockchain Patterns, Data Management Patterns, Security Patterns, Contract Structural Patterns.

**Module IV:** Quality Impact of Using Blockchain-On Chain Data Cost, Smart Contract Cost, Cost Models, Performance Characteristics of Blockchain, Architectural Performance Modelling, Predicting Latency for Blockchain-Based Systems, Confidentiality, Integrity, Safety, Maintainability, Availability and Reliability

**Module V:** Case Studies as Blockchain technology based Applications, Agri Digital-Agricultural Supply Chains, The Agri Digital Vision, Designing for a Business Use Case, SecureVote-The MVP Prototype, Building Tokenvote, Details and Code Samples

#### Textbooks:

1. Narayanan, J. Bonneau, E. Felten, A. Miller, and S. Goldfeder, “Bitcoin and Cryptocurrency Technologies”, Princeton University Press, 2016.
2. Xiwei Xu, I. Weber, M. Staples, “Architecture for Blockchain Applications”, Springer, 2018.

#### Reference Books:

1. M. Swan, “Blockchain: Blueprint for a New Economy”, OReilly, 2015.
2. Lecture Notes of S. Shukla (IIT Kanpur) , “Introduction to Block Chain Technology and Applications”.

Course Code	Course Name	L T P C
	AI for Cyber Security	3 0 0 3

**Objectives:** To develop expertise in applying artificial intelligence and machine learning techniques to cyber security problems, including threat detection, intrusion and anomaly detection, deep learning-based analysis, threat intelligence automation, and adversarial ML defenses.

#### Learning Outcomes:

Upon successful completion of this course, students will be able to:

1. Understand the role of AI in cybersecurity, compare AI-based and traditional security approaches, and apply core machine learning algorithms (supervised, unsupervised, reinforcement learning) for threat detection.
2. Design and implement anomaly-based intrusion detection systems using clustering and feature engineering, especially for encrypted network traffic.
3. Apply deep learning models (CNNs, RNNs, autoencoders) for malware classification, log analysis, and phishing URL detection.
4. Use AI techniques for network traffic analysis, malware and botnet behavior analysis, threat intelligence automation, NLP-based threat analysis, and AI-assisted incident response and threat hunting.
5. Analyze machine learning attacks such as backdoor and poisoning attacks, and evaluate adversarial threats and defenses in AI-driven cybersecurity systems

**Prerequisites:** Prior knowledge of Machine Learning basics and Web Security

#### Contents:

**Module I:** Introduction - Role of AI in Cyber Security, Overview of Threat Landscape, AI vs Traditional Security Approaches, AI Algorithms Overview: Supervised, Unsupervised, Reinforcement Learning, Tools & Frameworks (TensorFlow, Scikit-learn, Keras), Threat Detection-Classification Algorithms: Decision Trees, Random Forests, SVM,

**Module II:** Anomaly Detection using Clustering (K-Means, DBSCAN) , Feature Engineering for Threat Data , Intrusion Detection in Encrypted Traffic, Challenges in Intrusion Detection for Encrypted Traffic, Workflow of ML-IDS for Encrypted Traffic, ML-based Solutions for Encrypted Traffic, Case Study: ML in Intrusion Detection System (IDS)

**Module III:** Deep Learning for Cyber Security- Neural Networks and Deep Learning Basics, Convolutional Neural Networks (CNN) for Malware Classification, Recurrent Neural Networks (RNN) for Log Analysis,

Autoencoders for Anomaly Detection, Case Study: Deep Learning for Phishing URL Detection

**Module IV:** Network Traffic Analysis using AI, Malware Behaviour Analysis, Botnet Detection using AI, Use of Graph Neural Networks in Cyber Security, AI for Threat Intelligence Automation, Natural Language Processing (NLP) in Threat Intelligence, AI in Incident Response and Threat Hunting. Ethical AI Use and Adversarial ML in Cyber Security

**Module V:** Machine Learning Attacks - Threat Model and Targeted Models, Backdoor Attack Scenarios, Attackers' Capability, Attackers' Objective, Targeted ML Models, Attack Formulation and Challenges, Backdoor Attack Formulation, Challenges Poison-label Backdoor Attack, Backdoor Trigger Optimization, Clean-label Backdoor Trigger Design, Clean-label Backdoor Trigger Optimization, Evaluation

**Textbooks:**

1. Cyber Security Using Modern Technologies: Artificial Intelligence and the Internet of Things" Srinivas Mukkamala & Andrew S. Prowell Springer, 2021.
2. Introduction to Machine Learning with Applications in Information Security" Mark Stamp CRC Press, 2018.

**Reference Books:**

1. Anomaly Detection: A Survey" Chandola V., Banerjee A., & Kumar V ACM Computing Surveys.
2. Guide to Computer Network Security" Joseph Migga Kizza Springer.
3. Artificial Intelligence for Cybersecurity: Techniques, Tools, and Applications" James Ma Weiming Wiley, 2020.

Course Code	Course Name	L-T-P: C
	<b>Reinforcement Learning</b>	<b>3-0-2:4</b>

**Description:** This course focuses on designing computational agents that learn sequential, optimal decisions through trial-and-error by maximizing long-term reward. The curriculum covers mathematical foundations, including Markov Decision Processes, value functions, and the Bellman framework, progressing from tabular methods to function approximation and deep reinforcement learning. This course provides strong theoretical and practical foundations for modeling real-world decision problems and implementing algorithms for autonomous agents in

fields like artificial intelligence, robotics, and networking.

**Objectives:** Reinforcement Learning focuses on the design of computational agents that learn to make sequential decisions through interaction with an environment. The central theme of this course is learning through trial-and-error, where an agent improves its behavior by maximizing long-term cumulative reward. The course introduces the mathematical foundations of decision-making under uncertainty using Markov Decision Processes, value functions, and policy optimization methods. Emphasis is placed on understanding the Bellman framework, model-based and model-free learning, and the transition from tabular methods to function approximation and deep reinforcement learning. Students will learn to model real-world decision problems as reinforcement learning tasks and implement algorithms that enable autonomous agents to adapt, optimize, and act intelligently. The course aims to provide strong theoretical and practical foundations for advanced research and applications in artificial intelligence, robotics, control systems, networking, and autonomous systems.

**Learning Outcomes:** Learning Outcomes: On completion of this course the student should be able to:

1. Understand intelligent agents and decision-making frameworks.
2. Formulate problems as Markov Decision Processes (MDPs).
3. Implement classical RL algorithms.
4. Understand deep reinforcement learning.
5. Design RL agents for real-world domains (robotics, networking, telecom, games).
6. Explore multi-agent RL and policy learning frameworks.

**Contents:**

**Module 1:** Introduction to Intelligent Agents What is Intelligence? Rational Agents, Agent-Environment Interaction, Types of Agents: Simple Reflex, Model-Based, Goal-Based, Utility-Based, Learning Agents, Deterministic vs Stochastic Environments, Fully vs Partially Observable.

**Practical-1:** Agent-environment loop implementation in Python

**Module 2:** Trial-and-Error Learning Supervised vs Unsupervised vs Reinforcement Learning, Reward-based learning, Trial and Error concept, Exploration vs Exploitation dilemma, Multi-Armed Bandit problem,  $\epsilon$ -greedy strategy, UCB, Regret analysis (basic intuition)

**Practical-2:** Implement K-armed bandit, Compare  $\epsilon$ -greedy vs UCB, Plot cumulative reward and regret

**Module 3:** Markov Decision Processes (MDP Framework) Markov Property, MDP components: States (S), Actions (A), Transition Probability (P), Reward Function (R), Discount factor ( $\gamma$ ), Policy, Value Function, Bellman Equation, optimality condition

**Practical-3:** Implement GridWorld MDP, Solve environment manually, Visualize Value iteration convergence

**Module 4:** Model-Free Reinforcement Learning Monte Carlo Methods, Temporal Difference Learning, TD(0), SARSA, Q-Learning, On-policy vs Off-policy learning, Convergence intuition

**Practical-4:** Implement SARSA and Q-Learning, Compare performance.

**Module 5:** Function Approximation & Deep RL Large state space, Curse of dimensionality, Function approximation, Linear function approximation, Neural networks in RL, Deep Q Network (DQN), Experience replay, Target networks, Double DQN (conceptual), Policy Gradient, REINFORCE, Actor-Critic

**Practical-5:** Implement DQN using PyTorch for Train CartPole, Compare tabular vs DQN

**Module 6:** Multi-agent MDP, Cooperative vs Competitive, Game Theory basics, Nash Equilibrium (intuitive), Independent Q-Learning, Centralized Training Decentralized Execution (CTDE), MADDPG (conceptual), Various Applications and Usecases.

**Practical-6:** Two-agent gridworld, Cooperative navigation example.

**Text Book:**

1. Reinforcement Learning: An Introduction, Second Edition, Sutton and Barto MIT Press, 2018

**Reference Books:** Selected Papers on Reinforcement Learning, Multi-agent Reinforcement Learning.

Course Code	Course Name	L-T-P: C
	<b>Introduction to Robotics</b>	<b>3-0-2:4</b>

**Description:** This course introduces the fundamental concepts of robotics by integrating principles from mechanical systems, electronics, and computer science. It provides a comprehensive understanding of robot structure, sensing, actuation, and intelligent control. The course emphasizes both theoretical foundations and

practical insights relevant to modern robotic applications.

**Objectives:** The course aims to build a strong foundation in the core components and working principles of robotic systems. It seeks to familiarize learners with kinematics, sensing, and control strategies used in robotics. Additionally, the course prepares students to apply robotics concepts in real-world applications such as automation, autonomous systems, and intelligent machines.

**Learning Outcomes:** After completing the course, students will be able to:

1. Understand the basic architecture and components of robotic systems.
2. Analyze forward and inverse kinematics of robotic manipulators.
3. Apply concepts of sensors and actuators in robot design.
4. Develop simple control strategies for robotic motion.
5. Explore applications of robotics in industrial and autonomous domains

**Prerequisite:** None

**Contents:**

**Module I:** Introduction to Robotics and Components - History and evolution of robotics, classification of robots, robot anatomy and configurations, degrees of freedom, coordinate systems, applications of robotics

**Module II:** Kinematics of Robots - Rigid body transformations, homogeneous transformations, forward kinematics, inverse kinematics, Denavit-Hartenberg (D-H) parameters.

**Module III:** Sensors and Actuators - Types of sensors (proximity, vision, force, tactile), actuators (DC motors, stepper motors, servo motors), signal processing, interfacing basics.

**Module IV:** Robot Control and Applications - Control systems in robotics, trajectory planning, motion control, introduction to ROS, industrial automation, autonomous robots.

**Text Books:**

1. Craig, J. J. (2018). *Introduction to Robotics: Mechanics and Control* (4th ed.). Pearson Education.
2. Siciliano, B., Sciavicco, L., Villani, L., & Oriolo, G. (2019). *Robotics: Modelling, Planning and Control*. Springer.

**Reference Books:**

1. Corke, P. (2023). *Robotics, Vision and Control: Fundamental Algorithms in MATLAB* (3rd ed.). Springer.
2. Spong, M. W., Hutchinson, S., & Vidyasagar, M. (2020). *Robot Modeling and Control* (2nd ed.). Wiley.

Course Code	Course Name	L-T-P: C
	<b>Topics in Security and Privacy</b>	<b>3-0-2:4</b>

**Objectives:** The course develops a sound understanding of the mathematical preliminaries, namely number theory and algebra of finite fields, which are required to understand and design cryptographic protocols. It familiarizes the students with the notions and methods to ensure privacy preserving data publishing. Finally, the course aims to apply the conceptual understandings to the advanced applications like key management, digital cash, electronic elections etc.

#### Learning Outcomes

1. Understand and apply core cryptographic primitives.
2. Analyze advanced cryptographic protocols.
3. Design secure systems for data and computation outsourcing.
4. Apply privacy-preserving techniques in real-world scenarios.
5. Understand modern cryptographic applications and emerging paradigms.

#### Contents:

**Module I:** Preliminaries: Number theory and algebra, finite fields. Symmetric key encryption: Stream ciphers and block ciphers. Public key cryptography, digital signatures, attacks, hash functions, authentication schemes, key exchange algorithm, public key infrastructure.

**Module II:** Identification schemes, interactive proofs, commitment protocols, zero knowledge proofs, non-interactive proofs.

**Module III:** Secure outsourcing and sharing of data. secure key management and key distribution. Secure outsourcing of computations. Garbled circuit evaluation, multi-party computation.

**Module IV:** Privacy Regulations, Noiseless Differential Privacy, Privacy preserving Data Mining techniques. Privacy preserving data publishing. Fundamental Concepts, anonymization methods, privacy models, anonymization method for different data types eg.

transaction data, social networks data etc. One-Time Data Publishing, Multiple-Time Data Publishing.

**Module V:** Secret sharing schemes, digital cash, electronic voting, elliptic curve cryptosystems, identity-based encryption.

#### Text Books:

1. Jonathan Katz, Yehuda Lindell, Introduction to Modern Cryptography: Principles and Protocols, Chapman & Hall/CRC Cryptography and Network Security Series, 2007.
2. Neal Koblitz, "A Course in Number theory and cryptography", Springer, 2007.
3. Hans Delfs, Helmut Knebl, "Introduction to Cryptography: Principles and Applications", Springer.
4. Alfred J. Menezes, Paul C. van Oorschot, Scott A. Vanstone, "Handbook of Applied Cryptography", CRC Press, 1996.
5. Relevant Research Papers.

Course Code	Course Name	L T P: C
	<b>Time Series Analysis and Forecasting using Machine Learning and Deep Learning</b>	<b>3 0 0: 3</b>

**Objectives:** To provide students with a comprehensive understanding of time series data analysis and forecasting techniques, covering fundamental concepts, statistical and machine learning models, evaluation methodologies, and deep learning approaches, enabling them to design and implement effective forecasting solutions for real-world temporal data applications.

**Learning Outcomes:** Upon successful completion of this course, students will be able to:

1. Understand the fundamentals of time series data, including its components, characteristics, and differences from traditional machine learning data.
2. Apply preprocessing techniques such as decomposition, detrending, differencing, and transformation to prepare time series data for modeling.
3. Analyze and evaluate forecasting models using appropriate validation strategies, baseline methods, and performance metrics.
4. Develop and implement classical statistical models such as autoregressive, moving average, ARIMA, and seasonal ARIMA for time series forecasting.
5. Design and build machine learning and deep learning models, including regression, MLP, CNN, and LSTM architectures, for solving real-world time series forecasting problems.

**Course Outcomes:**

After completion of the course, students will be able to:

1. Understand fundamental concepts of time series data, components, and forecasting challenges.
2. Apply preprocessing techniques such as decomposition, detrending, and stationarity transformation.
3. Analyze and evaluate forecasting models using validation strategies and performance metrics.
4. Develop classical statistical models such as AR, MA, ARIMA, and SARIMA.
5. Design and implement machine learning and deep learning models for real-world time series forecasting problems.

**Pre Requisites:** Machine Learning

**Contents:****Module 1: Fundamentals of Time Series:**

Introduction to time series data including definitions, terminology, and real-world applications, understanding the components of time series such as trend, seasonality, cyclicity, and noise, visualization and exploratory analysis of temporal data, differences between time series data and traditional machine learning datasets, transformation of time series into supervised learning problems using sliding window techniques, handling univariate and multivariate time series, and approaches for single-step and multi-step forecasting.

**Module 2: Temporal Structure and Preprocessing:**

Concepts of white noise and random walk processes, understanding stationarity and its importance in time series modeling, decomposition of time series into additive and multiplicative components, techniques for removing trends including differencing and model-based approaches, handling seasonality through seasonal adjustment methods, data transformations such as scaling and logarithmic transformations to stabilize variance and improve model performance.

**Module 3: Forecast Evaluation and Problem Framing:**

Different forecasting strategies including one-step and multi-step forecasting as well as direct and recursive approaches, backtesting methodologies such as train-test split and rolling window validation, evaluation metrics for forecasting performance including MAE, RMSE, MAPE, and related measures, development of baseline models such as naive, seasonal naive, and persistence models, analysis of residual errors and diagnostic checking, and reframing forecasting problems into regression and classification settings for machine learning applications.

**Module 4: Classical Forecasting Models:**

Autoregressive and moving average modeling

approaches for time series forecasting, integration of models into ARMA and ARIMA frameworks, extension to seasonal models such as SARIMA, application of the Box-Jenkins methodology including model identification, parameter estimation, and diagnostic validation, use of autocorrelation and partial autocorrelation for model selection, hyperparameter tuning techniques including grid search, and estimation of forecast intervals along with uncertainty analysis.

**Module 5: Machine Learning and Deep Learning for Time Series:**

Framing time series forecasting problems for machine learning models, feature engineering techniques for temporal data, application of machine learning algorithms such as linear regression, decision trees, and ensemble methods, deep learning approaches including multilayer perceptrons, convolutional neural networks, and recurrent neural networks such as LSTM, model optimization and regularization strategies, handling multivariate time series data, and exploration of real-world applications including energy forecasting, financial time series prediction, and healthcare signal analysis.

**Textbook:**

1. Forecasting: Principles and Practice, Hyndman, R. J., & Athanasopoulos, G., *Forecasting: Principles and Practice* (3rd Edition), OTexts.

**Reference Books:**

1. Forecasting Methods and Applications, Makridakis, S., Wheelwright, S. C., & Hyndman, R. J., *Forecasting Methods and Applications* (3rd Edition), Wiley.
2. Time Series Analysis and Its Applications, Shumway, R. H., & Stoffer, D. S., *Time Series Analysis and Its Applications* (4th Edition), Springer.
3. Deep Learning for Time Series Forecasting, Brownlee, J., *Deep Learning for Time Series Forecasting*, Machine Learning Mastery.

भारतीय  
सूचना प्रौद्योगिकी  
संस्थान वडोदरा



Indian  
Institute of Information  
Technology Vadodara

(शिक्षा मंत्रालय, भारत सरकार के अधीन राष्ट्रीय महत्व का संस्थान)

(An Institution of National Importance under Ministry of Education, Govt. of India)

॥ नवोन्मेषाय प्रौद्योगिकी, लोकहिताय प्रयोजनम् ॥



# B.TECH. CURRICULUM - 2026

Indian Institute of Information Technology Vadodara  
Current Address: c/o Block No.9, Government Engineering College, Sector 28,  
Gandhinagar, Gujarat. Pin: 382028

Indian Institute of Information Technology Vadodara - International Campus Diu,  
Address: Education Hub, Kevdi - Diu (U.T).  
Pin: 362520