

# Curriculum-2018

## Bachelor of Technology

in

Computer Science and Engineering

&

Information Technology



**Indian Institute of Information Technology Vadodara**

**Table of Contents**

	<b>Page No.</b>
<b>B. Tech. Program</b>	<b>1-3</b>
<b>B. Tech. Program: Course Categories and Distribution of Credits</b>	<b>4-7</b>
<b>Courses Structure and Contents:</b>	
<b>First Year (Semester-I)</b>	<b>8-13</b>
<b>First Year (Semester-II)</b>	<b>14-18</b>
<b>Second Year (Semester-III)</b>	<b>19-25</b>
<b>Second Year (Semester-IV)</b>	<b>26-31</b>
<b>Third Year (Semester-V)</b>	<b>32-37</b>
<b>Third Year (Semester-VI)</b>	<b>38-41</b>
<b>Fourth Year (Semester-VII)</b>	<b>42-45</b>
<b>Fourth Year (Semester-VIII)</b>	<b>46-47</b>
<b>Appendix-I: List of Program Electives for CSE/IT Program</b>	<b>48</b>
<b>Appendix-II: List of Electives from other branch of Engineering</b>	<b>48</b>
<b>Appendix-III: List of Open Electives</b>	<b>48</b>

## B. Tech. Program

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

1. Computer Science and Engineering (CSE)
2. Information Technology (IT)

### 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.

### 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 be  $U_1, U_2, \dots$  and the numeric values of the corresponding grade awarded in the courses be  $G_1, G_2, \dots$ , 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

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

#### 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** shall be awarded to graduate(s) meeting the following criteria:

1. Secured the highest *CPI* in the admission batch (across departments/ branches),
2. Secured minimum *CPI* of 9.00,
3. Not been put on academic probation during the academic program duration,
4. Not been put on disciplinary probation during the academic program duration and
5. No fail 'F' grade in the transcript.

In case of more than one candidates qualify for the award, all candidates will be awarded the medals.

#### **[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. Program

### Course Categories and Distribution of Credits

#### 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

#### Range of Credits for B. Tech. degree

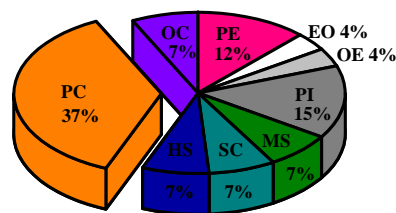
For a student to be eligible to get B. Tech. degree, a range of credits earned should be in between 160-180.

#### Course Categories and Distribution of Credits

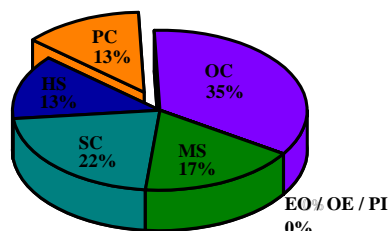
Code	Course Categories	Credits
MS	Mathematics & Statistics	12
SC	Natural Science (Physics, Chemistry, Biology), Environmental Science	12
HS	Humanities, Social Science, Literature, Management & Soft-skills (including Electives)	12
PC	Core Engineering Courses (from the branch of study)	60
OC	Core Engineering Courses (from the other branch of Engineering)	12
PE	Program Elective (from the branch of study)	20
EO	Elective (from other branch of Engineering)	6
OE	Open Elective (Science / Humanities / Engineering: not more than one from each )	6
PI	Projects, Internships (Research/Industrial)	24
<b>Total Credits</b>		<b>164*</b>

\*Minor variation is allowed as per need of the respective disciplines.

B. Tech. Program: Distribution of Credits (in %)



Distribution of Credits in 1<sup>st</sup> Year (in %)



#### Distribution of Credits in 1<sup>st</sup> year of B. Tech. (CSE/IT) Program:

Code	Course Categories	Credits
MS	Mathematics & Statistics	8
SC	Basic Sciences (Physics, Chemistry, Biology), Environmental Science.	10
HS	Humanities, Social Science, Literature, Management, Soft-skills	6
PC	Core Engineering Courses (from the branch of study)	6
OC	Core Engineering Courses (from the other branches of Engineering)	16
PE	Program Electives (from the branch of study)	0
EO	Electives (from other branches of Engineering)	0
OE	Open Electives (Science / Humanities / Engineering: Not more than one from each)	0
PI	Projects, Internships (Research/Industrial)	0
<b>Total Credits</b>		<b>46</b>

**Distribution of Credits in 2<sup>nd</sup> to 4<sup>th</sup> years of B. Tech. (CSE) Program:**

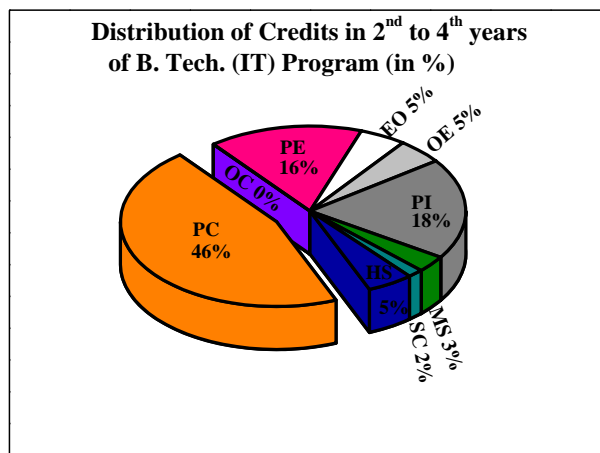
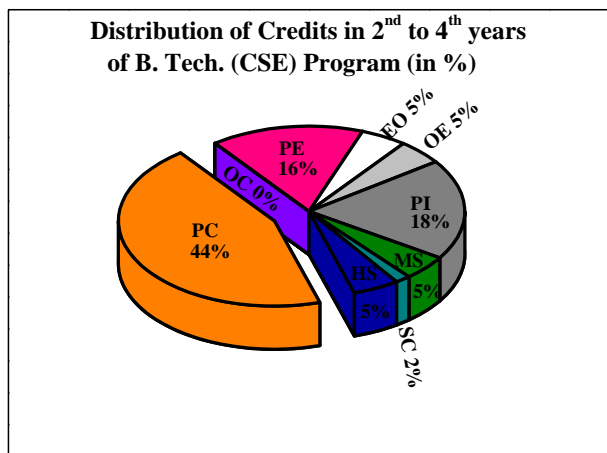
Code	Course Categories	Credits
MS	Mathematics & Statistics	6
SC	Basic Sciences (Physics, Chemistry, Biology), Environmental Science.	2
HS	Humanities, Social Science, Literature, Management, Soft-skills	6
PC	Core Engineering Courses (from the branch of study)	56
OC	Core Engineering Courses (from the other branches of Engineering)	0
PE	Program Electives (from the branch of study)	20
EO	Electives (from other branches of Engineering)	6
OE	Open Electives (Science / Humanities / Engineering: Not more than one from each)	6
PI	Projects, Internships (Research/Industrial)	24
<b>Total Credits*</b>		<b>126</b>

\*Total Credits for Degree is 172.

**Distribution of Credits in 2<sup>nd</sup> to 4<sup>th</sup> years of B. Tech. (IT) Program:**

Code	Course Categories	Credits
MS	Mathematics & Statistics	4
SC	Basic Sciences (Physics, Chemistry, Biology), Environmental Science.	2
HS	Humanities, Social Science, Literature, Management, Soft-skills	6
PC	Core Engineering Courses (from the branch of study)	57
OC	Core Engineering Courses (from the other branches of Engineering)	0
PE	Program Electives (from the branch of study)	20
EO	Electives (from other branches of Engineering)	6
OE	Open Electives (Science / Humanities / Engineering: Not more than one from each)	6
PI	Projects, Internships (Research/Industrial)	24
<b>Total Credits*</b>		<b>125</b>

\*Total Credits for Degree is 171.



Semester wise distribution of Credits:

A. CSE Branch

	Cr.	SEM-I			SEM-II			SEM-III			SEM-IV			SEM-V			SEM-VI			SEM-VII			SEM-VIII										
		L	T	P	C	L	T	P	C	L	T	P	C	L	T	P	C	L	T	P	C	L	T	P	C								
MS	14	3	1	0	4	3	1	0	4	3	1	0	4	0	1	2	2																
					4				4				4				2			0			0			0							
SC	12	3	1	2	5	3	1	2	5	2	0	0	2																				
					5				5				2			0			0			0			0								
HS	12	2	0	2	3	3	0	0	3	1	1	2	3	3	0	0	3																
					3				3				3			0		0			0			0	0								
PC	61					3	1	2	5	3	0	3	5	3	0	3	5	3	0	2	4	3	0	2	4	3	0	2	4				
										3	0	2	4	3	0	3	5	3	0	3	5	3	0	2	4								
										3	0	3	5	3	0	2	4	3	0	2	4												
														3	0	0	3																
					0				5				14				17			13			8			4			0				
PE	20													3	0	2	4	3	0	2	4	3	0	0	3								
														3	0	0	3	0	1	2	2	3	0	2	4								
					0				0				0				7			6			7			0							
OC	17	3	0	3	5	3	1	3	6																								
		3	1	3	6																												
					11				6			0		0		0		0			0			0		0							
EO	6																	3	0	0	3	3	0	0	3								
					0				0			0		0		0		0			3			3			0						
OE	6																		3	0	0	3	3	0	0	3							
					0				0			0		0		0		0			3			3			0						
PI	24													0	0	0	3					0	0	0	3	0	0	36	18				
					0				0			0		0		3								3			36	18					
	172																																
Tot	172	14	3	10	23	15	4	7	23	15	2	10	23	15	1	10	22	15	0	9	23	15	1	8	20	15	0	4	20	0	0	36	18
Contact hrs. per week					27				26				27				26				24				24				19			36	



**B. IT Branch**

	Cr.	SEM-I			SEM-II			SEM-III			SEM-IV			SEM-V			SEM-VI			SEM-VII			SEM-VIII										
		L	T	P	C	L	T	P	C	L	T	P	C	L	T	P	C	L	T	P	C	L	T	P	C								
MS	12	3	1	0	4	3	1	0	4	3	1	0	4																				
					4				4				4				0			0			0			0							
SC	12	3	1	2	5	3	1	2	5	2	0	0	2																				
					5				5				2				0			0			0			0							
HS	12	2	0	2	3	3	0	0	3	1	1	2	3	3	0	0	3																
					3				3				3				3			0			0			0							
PC	62	3	0	3	5				3	0	3	5	3	0	3	5	3	0	2	4	3	0	2	4									
									3	0	2	4	3	0	3	5	3	0	2	4	3	0	2	4									
									3	0	3	5	3	0	2	4	3	0	2	4													
													3	0	2	4																	
					5				0				14				18			13			8			4	0						
PE	20													3	0	2	4	3	0	2	4	3	0	0	3								
														3	0	0	3	0	1	2	2	3	0	2	4								
					0				0				0				7			6			7			0							
OC	17	3	1	3	6	3	1	3	6																								
						3	1	2	5																								
EO	6																	3	0	0	3	3	0	0	3								
					0				0				0				0			3			3			0							
OE	6																	3	0	0	3	3	0	0	3								
					0				0				0				0			3			3			0							
PI	24													0	0	0	3					0	0	0	3	0	0	36	18				
					0				0				0				3						3			3		18					
	171																																
Tot	171	14	3	10	23	15	4	7	23	15	2	10	23	15	0	10	21	15	0	9	23	15	1	8	20	15	0	4	20	0	0	36	18
Contact hrs. per week					27				26				27				25				24				24			19			36		

**Course Structure: First Year**

(Common to all branch of studies)

**Semester-I**

Course Code	Course Name	L	T	P	C
MA101	Mathematics-I (Linear Algebra and Matrices)	3	1	0	4
PH100*	Mechanics and Thermodynamics	3	1	0	4
PH160*	Mechanics and Thermodynamics Lab	0	0	2	1
IT101	Computer Programming and Problem Solving	3	0	0	3
IT161	Computer Programming and Problem Solving Lab	0	0	3	2
EC100*	Basic Electronic Circuits	3	1	0	4
EC160*	Basic Electronic Circuits Lab	0	0	3	2
HS101	Spoken and Written Communication	2	0	2	3
<b>Total</b>		<b>14</b>	<b>3</b>	<b>10</b>	<b>23</b>

\*The Course may be offered in both Autumn and Winter Semesters.

**Semester-II**

Course Code	Course Name	L	T	P	C
MA102	Mathematics-II (Discrete Mathematics)	3	1	0	4
PH110*	Waves and Electromagnetics	3	1	0	4
PH170*	Waves and Electromagnetics Lab	0	0	2	1
EE100*	Basic Electrical Engineering	3	1	0	4
EE160*	Basic Electrical Engineering Lab	0	0	3	2
CS102	Introduction to Data Structures	3	0	0	3
CS162	Introduction to Data Structures Lab	0	1	2	2
HS102	Science Technology and Society	3	0	0	3
<b>Total</b>		<b>15</b>	<b>4</b>	<b>7</b>	<b>23</b>

\*The Course may be offered in both Autumn and Winter Semesters.

**Semester-I: Courses and Contents**

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

**Description:** The course aims at introducing the fundamental concepts of linear algebra culminating in abstract vector spaces and linear transformations. The course starts with systems of linear equations, matrices, matrix algebra, eigenvalues and eigenvectors. The course then goes on to introduce some basic concepts of the theory of vector spaces in the concrete setting of real  $n$ -dimensional vector space and linear transformations. At the end, numerical methods are introduced to apply the theory to real life problems. The subject material is of vital importance in all fields of mathematics and sciences in general.

**Objective:** Students will be able to apply the concepts and methods described in the syllabus and solve problems using linear algebra.

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

1. Solve the system of linear equations and apply the theory of vector spaces to understand it.
2. Solve large system of linear equations using computational techniques.
3. Demonstrate competence with the basic ideas of linear algebra including concepts of linear systems, independence, theory of matrices, linear transformations, bases and dimension, eigenvalues, eigenvectors and diagonalization.

**Prerequisite:** None

**Contents:**

**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:  $Ax=b$

**Canonical Factorizations:** Eigenvalues and Eigenvectors, Companion Matrices and Characteristic Polynomial, Method of Danilevsky for Characteristic Polynomial, diagonalization-Matrices with a Full-Set of Eigenvectors, The Cayley-Hamilton Theorem, Triangulization and Unitary Diagonalization of a Matrix, Schur's Lemma and the Spectral Theorem, QR-Decomposition, QR-Algorithm for Hessenberg Matrices, Singular Value Decomposition.

**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, Canonical forms, Rank of a Linear Transformation.

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

**Text Book:**

1. *Introduction to Linear Algebra*, Gilbert Strang, 5<sup>th</sup> Ed, SIAM, 2016.

**Reference Books:**

1. *Linear Algebra*, Kunze Ray, Hoffman Kenneth 2<sup>nd</sup> Ed, Phi Learning, 2014.
2. *Fundamentals of Matrix Computations*, David S. Watkins, 3<sup>rd</sup> ed, Wiley.

Course Code	Course Name	L-T-P: C
PH100	Mechanics and Thermodynamics	3-1-0: 4

**Objectives:** The course provides engineering students with important foundational knowledge about mechanics, and thermodynamics and its application to common engineering systems.

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

1. Demonstrate knowledge of the physical principles that describe mechanics, materials, heat transfer, and thermodynamics;
2. Apply physics principles to common physical systems;
3. Use the methods of algebra, vectors and calculus in quantitative and qualitative predictions of behavior of physical systems and
4. Associate the correct unit with different physical quantity they use.

**Prerequisite:** None

**Contents:**

**Mechanics:**

**Review of Newtonian Mechanics:** Vectors and their time derivatives, Inertial and non-inertial frames of reference, Centrifugal and Coriolis forces; Work-Energy Theorem; Conservation Principles, Collision problem in laboratory and center of mass frame, Motion under Central Force and its universal features, Oscillatory Motion-Free, Damped and Driven Oscillations.

**Introduction to Quantum Mechanics:** Double-slit experiment, de Broglie's hypothesis, Uncertainty Principle, Wave-Function and Wave-Packets, Phase and Group-velocities, Schrödinger Equation, Probabilities and Normalization, Expectation values, Eigenvalues and Eigen-functions, Applications of

Schrödinger Equation: Particle in a box, Finite Potential well, Harmonic oscillator, Hydrogen Atom problem.

**Thermodynamics:**

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.

**Text Books:**

1. *An Introduction to Mechanics*; D. Kleppner and R. Kolenkow, Second Edition.
2. *Concepts of Modern Physics*; Arthur Beiser, Sixth Edition.
3. *Heat and Thermodynamics*; M. W. Zemansky and R. H. Dittman, Seventh Edition.

**Reference Book:**

1. *The Feynman Lectures on Physics, Vol-I & III*, Feynman, Leighton and Sands; Pearson Education.

Course Code	Course Name	L-T-P: C
PH160	Mechanics and Thermodynamics Lab	0-0-2: 1

**Objectives:** The course provides a list of experiments to familiarize students with the basic experiments of mechanics and thermodynamics.

**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. Heat capacity of gases, specific heat of solids and the laws of thermodynamics.

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.

**Prerequisite:** None

**List of Experiments:**

1. Motion on Linear Air-Track (Virtually Frictionless Surface).
2. Centripetal Force and Moment of Inertia.
3. Damped oscillation with spring-mass system.

4. Oscillations in Coupled Pendulum.
5. Photoelectric effect and Determination of Planck's constant.
6. Franck Hertz Experiment.
7. Balmer Series and Determination of Rydberg constant.
8. Determination of Heat capacity of gases.
9. Determination of Specific Heat of Solids.
10. Mechanical Equivalent of Heat.

**Reference:** Laboratory Manual.

Course Code	Course Name	L-T-P: C
IT101	Computer Programming and Problem Solving	3-0-0: 3

**Objectives:** The course provides concepts of computer programming and its roles in problem solving. It also introduces how to develop well-structured programs.

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

1. Understand the basic computing environment, limitations, and usage.
2. Understand basic problem solving and programming concepts.
3. Visualize and model the basic real-life problems into computational problems.
4. Analyze a given computational problem and convert it into an algorithm and flowchart.

**Prerequisite:** None

**Contents:**

**Introduction to Computers:** Computer Systems, Computing Environments.

**Introduction to Programming:** Programming methods, paradigms, problem solving techniques, algorithm development, flow charts, Editor, compiler, debugger, Software development.

**Basics of Procedural Programming:** Constants, variables, expressions, operators, assignment, basic input and output, built-in functions, program debugging.

**Variables and Operators:** Basic data types, precedence and order of evaluation, pointers, memory allocation of variables.

**Control Structures:** Selection statements, iteration statements.

**Functions and Program Structure:** Return values, actual and formal parameters, parameter passing: call by value versus call by reference, external variables, scope rules, header files, and recursion.

**Arrays:** Character arrays, one and two dimensional arrays; pointer arrays, command-line arguments.

**I/O:** ASCII data files, file pointers, end-of-file.

**Basic Data Structures:** Structures, defining new types, enumerations, dynamic memory allocation, dynamic arrays, linked lists and other pointer-based structures.

**Text Book:**

1. *C How to Program*, 7th Ed., P Deitel and H Deitel, Prentice Hall of India, 2012.

**Reference Books:**

1. *C programming language*, 2nd Ed., Kernighan, Brian W. & Ritchie, Dennis M, New Delhi. Prentice Hall of India, 1998.
2. *A Structured Programming Approach Using C*, 1st Ed., Forouzan, Behrouz, Course Technology, 2012.
3. *Practical C Programming*, 3rd Ed, Oualline, Steve, Shroff Publishers, 2000.
4. *Programming In ANSI C* by E. Balagurusamy.

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

**Objectives:** The course provides a platform to enhance the analyzing and problem solving skills and learn to implement a list of programs in C or Python programming language.

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

1. Given a problem, identify and model the computational task involved.
2. Select a programming environment and convert the algorithm (or flowchart) into a program.
3. Choose the relevant data representation format based on the requirements of the problem.
4. Execute the program in the given environment. Understand the type of errors evolved if any. Produce convincing arguments to resolve the issues. Analyze the program execution environment.
5. Work as a team member among your peers.

**Prerequisite:** None

**List of Laboratory Assignments:**

**Part A (10 weeks):**

1. Program to find area and circumference of circle.
2. Program to convert temperature from degree centigrade to Fahrenheit.
3. Program to calculate sum of 5 subjects and find percentage.
4. Program to show swap of two no's without using third variable.
5. Program to reverse the digits of a given number.
6. Program to print a table of any number.
7. Program to find greatest in 3 numbers.
8. Program to find that entered year is leap year or not.
9. Program to shift input data by two bits to the left.

10. Program to display arithmetic operator using switch case.
11. Program to print stars Sequences (right triangular, Isosceles triangle, etc.).
12. Program to print Fibonacci series up to 100.
13. Program to find factorial of a number.
14. Program to find whether given no. is a prime no. or not.
15. Program to add two number using pointers.
16. Program to find the largest number in an array.
17. Program for removing the duplicate element in an array.
18. Program to add two matrices.
19. Program to multiply two matrices.
20. Program to find transpose of a matrix.
21. Program to swap two numbers using functions.
22. Program to show call by reference.
23. Program to find whether a string is palindrome or not.
24. Program to find occurrences of vowels, consonants, words, spaces and special characters in the given statement.
25. Program to create enumerated data type for 12 months. Display their values in integer constants.
26. Program for linear and binary search.
27. Program for bubble sort and insertion sort.
28. Program that would sort a list of names in alphabetical order.
29. Program to use (++/-- ) operator with return value of function.
30. Program to read characters from a text file and print number of vowels, consonants and other characters in the file. Assume that the file will consist of mostly English-language letters.

**Part B (4-6 weeks)**

A small project will be given in groups (at most 4 persons in each group). The objective is to apply knowledge of programming language primitives such as functions, structures and/or files in day-to-day applications.

**Reference:** Laboratory Instructions and handouts.

Course Code	Course Name	L-T-P: C
EC100	Basic Electronic Circuits	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 helps to develop the ability of analyzing actual electronic circuits that implements 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:**

**Introduction to Passive Circuit Elements & Sources:** Resistor, Capacitor, Inductor, Voltage and Current sources, Controlled Sources, Thevenin and Norton Theorem.

**Basics of Semiconductors:** Semiconducting Materials, Intrinsic and Extrinsic Semiconductors, Charge-carrier Density and Distribution, Fermi level.

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

**Bipolar Junction Transistors (BJTs):** BJT structure, Basic BJT operation mechanism, Input and Output characteristics of common-emitter configuration, Transistor Bias Circuits-Base Bias, Emitter Bias, Voltage-Divider Bias, Emitter Feedback Bias, Collector Feedback Bias, Emitter-Collector Feedback Bias, ac Models, Voltage Amplifiers, Common Collector and Common Base Amplifiers, Power Amplifiers, and Frequency Response.

**Field Effect Transistors:** JFETs-Device structure, Drain Curves, Transconductance Curve, Biasing Circuits, JFET Amplifiers, MOSFETs-Device structure, Depletion-Mode MOSFET, D-MOSFET Curves, Amplifiers, Enhancement-Mode MOSFET, Digital Switching, CMOS, Power FETs.

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

**Oscillators:** Amplifier with feedback, Condition of harmonic oscillation, Wein Bridge Oscillator, RC Oscillators, Colpitts Oscillators, 555 Timer and Circuits.

**Text Book:**

1. *Electronic Principles*, 7<sup>th</sup> Ed, Albert Malvino, and David Bates, Tata McGraw-Hill, 2007.

**Reference Books:**

1. *Electronic Devices*; 9<sup>th</sup> Edition, Thomas L. Floyd, Pearson.
2. *Microelectronic Circuits: Theory and Applications*, A.S. Sedra and K.C. Smith, Oxford University Press, Sixth Edition.

Course Code	Course Name	L-T-P: C
EC160	Basic Electronic Circuits Lab	0-0-3: 2

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

**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.

**Prerequisite:** None

**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. Design a regulated power supply using Zener diode and verify its characteristics.
5. Input and output current-voltage characteristics of  $n-p-n$  bipolar junction transistors in common-emitter configuration and determine transistor parameters.
6. Design a Common-Emitter transistor ( $n-p-n$ ) amplifier circuit, obtain the frequency response curve of the amplifier and determine the mid-frequency gain,  $A_{mid}$ , lower and higher cut-off frequencies.
7. Design Inverting and Non-Inverting Operational Amplifiers for a given specification.
8. Design two stage RC coupled common emitter transistor ( $n-p-n$ ) amplifier circuit and determine its frequency response curve.
9. Design Integrator and Differentiator using Operational Amplifier.
10. Design an active low pass and high pass filter of different orders using Op-Amplifier.

**Reference:** Laboratory Manual.

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

**Objectives:** This course aims to equip students with the ability to express their thoughts effectively and in an

accurate and precise manner by developing their oral and written skills in the English language. The course intends to expose the learner to various components of effective communication in the English language in order to provide a comprehensive understanding of what constitutes a communicative act.

**Course Outcomes:** At the end of the course, the students shall be able to acquire the following skills:

1. *Spoken Communication:* Develop the ability to clearly and precisely state ideas, questions, arguments and other communication ranging from informal discussion to formal presentations. Develop the ability to participate in group communication through the ability to condense large amount of information into a concise speech act. Learn and understand the dynamics of effective turn-taking and listening in order to engage in fruitful and constructive oral communicative acts. Developing focused reading skills.
2. *Written Communication:* Develop the ability to reduce thought to writing by effectively condensing large amounts of information. Acquire the skill of effective listening strategies in order to capture and demonstrate the core ideas through written mode of communication. Develop the faculty for critical thinking so as to enhance written English skills.

**Prerequisite:** None

**Contents:**

**Elements of Communication:** Course instructor should make the students aware of the elements of communication, the role of English language for effective communication, the process of communication and factors that influence communication (sender, receiver, channel, code, topic, message, context, feedback, noise, filters & barriers), importance of audience and purpose, the information gap principle, verbal and non-verbal communication: body language, general communication and business communication. An ability to communicate well is a key soft skill. Many other skills depend on good communication skills. By learning the processes involved in communication, the students shall be able to appreciate the importance of good communication skills in becoming a successful professional.

**Articulatory Phonetics:** Course instructor should introduce the students to the science of Articulatory Phonetics. It would be a basic training for mastering English sound system, particularly putting emphasis on British English, as well as basic knowledge on the Phonetics of English language. The focus shall be on IPA (International Phonetic Alphabet) symbols, the anatomy of speech organs, production and organization of speech sounds and phonetic transcriptions. Understanding the phonetics of English shall help students in using

dictionaries effectively and pronouncing words correctly.

**Multimedia and Discursive Communication-I:**

Course Instructor should make an optimal use of cinema for increasing the students' familiarity with English. Testing be done on the basis of the student's comprehension of the plot and their ability to describe scenes from the film. Classroom exercise of asking students to comment on the plot or scenes of a given film – not in writing but by standing before the entire class and speaking in English — be frequently carried out. The aim of this unit is to make the student feel confident about her/his ability to form sentence in English for discursive communication.

**Multimedia and Discursive Communication-II:**

Course Instructor should use audio tapes, Ted Lectures, radio news broadcast or celebrated speeches, etc. for exposing the students' to a real time' and good spoken English. Class room tests be set to check the students' ability to respond to their listening experience in writing. This will help the Course Instructor to continually assess the requirements of the students and provide corrective advice. Testing the writing skills of students will require setting several questions of very short composition tasks, from 50 words to 150 words. The topics chosen for the composition tasks should be selected from the topics covered in the classroom discussions or from the life on the campus.

**Literature and Communication Skills:** Students should be provided four to five extended samples of written English such as short stories or newspaper editorials for them to mark their difficulties – words, idioms, sentence structures, etc. This will help the students in improving their ability to do focused reading of serious written literature. Testing of the reading comprehension skills be tested by giving them in advance of the test several passages for reading. The Course Instructor may select one or more of those seen passages' for the examination purpose.

**Text Books:**

1. *Prism: Spoken and Written Communication, Prose & Poetry* published by Orient Longman, 2008.
2. *Technical Communication: Principles and Practice*, Second Edition by Meenakshi Raman and Sangeeta Sharma, Oxford Publications, 2009.

**Recommended Books, Essays and Short Stories:**

1. *English and Communication Skills for Students of Science and Engineering*. Dhanavel, S.P. Units 1-5, Chennai: Orient, Blackswan Ltd., 2009.
2. *Scientific English: A Guide for Scientists and Other Professionals*. 2 nd ed. Day, R A., Hyderabad: Universities Press, 2000.
3. *A Course in Phonetics*, Fifth Edition, Ladefoged, Peter (Harcourt, Brace, Jovanovich: Fort Worth), 2006.

4. Of Ambition- Francis Bacon
5. Of Innovations- Francis Bacon
6. With the Photographer – Stephen Leacock
7. Speech on Indian Independence – Jawaharlal Nehru
8. Socrates and the Schoolmaster – F. L. Brayne
9. The Bet – Anton Chekov
10. An Astrologer's Day – R. K. Narayan
11. The Gift of the Magi – O' Henry
12. The Monkey's Paw- W.W. Jacobs

**Semester-II: Courses and Contents**

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

**Description:** This course covers the basic concepts of discrete mathematics used in computer science that involve formal reasoning. It also includes counting, relations, graphs, trees, and number theory.

**Objective:** To develop logical thinking and its application to computer science and understand the basic principles of sets, functions and counting. The subject enhances one's ability to reason and ability to present a coherent and mathematically accurate argument.

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

1. Write valid arguments using logical notation,
2. Understand the basic principles of sets and relations, counting techniques and graph theory.
3. Acquire ability to describe and validate computer programs in a formal mathematical manner.

**Prerequisite:** None

**Contents:**

**Foundation:** Propositional and predicate logic, logical equivalences, predicates and quantifiers, translation from language to logical expressions, nested quantifiers, set theory, set operations, set identities and functions, inverse and composition functions, graph of functions.

**Number Theory:** Division operator, prime factorization, properties of prime numbers, prime number theorem, GCD and LCM, modular arithmetic and applications, sequences and summations.

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

**Mathematical Reasoning and Induction:** Rules of inference, direct proof, proof by contradiction, proof by contrapositive, mathematical induction and second law of mathematical induction.

**Recursion:** Definition, recursive algorithm, recurrence relations, solving recurrence relations.

**Relations:** Relations and their properties, applications and representations, equivalence relations, partial ordering, Hasse diagram.

**Graphs:** Introduction and terminology, representation, isomorphism, connectivity, Warshall's algorithm, Euler and Hamilton path, shortest path.

**Text Book:**

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. *Discrete Mathematics*, Richard Johnsonbaugh, Prentice Hall, 2007.
3. *Mathematics: A Discrete Introduction*, 3rd Ed., Edward R. Scheinerman, Cengage Learning, 2006.
4. *Mathematical Structure for Computer Science*, 6th Ed, J. Gersting, Freeman, 2006.

Course Code	Course Name	L-T-P: C
PH110	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:**

**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.

**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.

**Magnetostatics:** Magnetic Fields and Forces, Biot-Savart law and Ampere's law, Magnetic Vector Potential, Magnetization- Diamagnetism, Paramagnetism and Ferromagnetism, Ampere's Law in Magnetized



Materials-Auxiliary Field H, Magnetic permeability and susceptibility.

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

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

**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
PH170	Waves and Electromagnetics Lab	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.

**Prerequisite:** None

**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
EE100	Basic Electrical Engineering	3-1-0: 4

**Objectives:** The course introduces basics of electric and magnetic circuits, single-and three-phase ac circuits. It describes the principle of operation of electrical machines and let a student to identify the type of electrical machines for a given application. The course also introduces the power generation, transmission, distribution and power converters.

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

1. Analyze basic electric and magnetic circuits.
2. Explain working principles of electrical machines and power converters.
3. Recognize the ratings of different electrical apparatus.
4. Comprehend the power generation, transmission and distribution.

**Prerequisite:** EC100

**Contents:**

**AC Circuits:** Representation of sinusoidal waveforms, peak and rms values, phasor representation, real power, reactive power, apparent power, power factor. Analysis of single-phase ac circuits consisting of R, L, C, RL, RC, RLC combinations (series and parallel), resonance. Three-phase balanced circuits, voltage and current relations in star and delta connections.

**Transformers:** Ideal and practical transformer, equivalent circuit, losses in transformers, regulation and efficiency. Auto-transformer and three-phase transformer connections.

**Electrical Machines:** Single-phase induction motor: Construction, working, torque-speed characteristic and speed control of separately excited dc motor. Construction and working of a three-phase induction motor, Significance of torque-slip characteristic. Loss

components and efficiency, starting and speed control of induction motor. Construction and working of synchronous generators.

**Power Systems:** Power generation techniques, Transmission, Distribution, Grid, and Cost of Electricity.

**Power Converters:** DC-DC buck and boost converters, duty ratio control. Single-phase and three-phase voltage source inverters; sinusoidal modulation.

**Text Books:**

1. S. J. Chapman, “*Electric Machinery Fundamentals*”, 4th Edition, McGraw Hill Education, 2005.
2. D. P. Kothari and I. J. Nagrath, “*Basic Electrical Engineering*”, 3rd Edition, Tata McGraw Hill, 2010.

**Reference Books:**

1. D. P. Kothari and I. J. Nagrath, “*Electric Machines*”, 4th Edition, Tata McGraw Hill, 2010.
2. D. C. Kulshreshtha, “*Basic Electrical Engineering*”, McGraw Hill, 2009.
3. L. S. Bobrow, “*Fundamentals of Electrical Engineering*”, Oxford University Press, 2011.
4. E. Hughes, “*Electrical and Electronics Technology*”, Pearson, 2010.

Course Code	Course Name	L-T-P: C
EE160	Basic Electrical Engineering Lab	0-0-3: 2

**Objectives:** The lab course provides hands-on experience with single- and three-phase electric circuits, and electrical machines such as transformers, motors and generators.

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

1. Connect and analyze single- and three-phase electric circuits.
2. Distinguish between star- and delta-connections.
3. Understand transformer action.
4. Understand performance characteristics of electric motors and generators.

**Prerequisite:** EC160

**List of Experiments:**

1. The magnetization characteristics of ferromagnetic material of two coils connecting them in series, parallel and to determine their effective mutual inductance.
2. Determine the real and reactive powers in a load consisting of resistor and inductor. Improve the power factor by connecting an appropriate capacitor in parallel to the load.
3. The load regulation and efficiency characteristics of the given single phase Transformer by direct loading

method and to learn the parallel operation of the Transformers and their load sharing behavior.

4. Determine the three phase power consumed using 2-wattmeters for different types of connections (star or delta) and loadings (balanced and unbalanced).
5. Perform load test on the induction motor by loading the motor using mechanical loading arrangement. Plot torque vs slip (%) and performance characteristics vs output power.
6. To conduct an Open circuit test on a separately excited dc generator and to determine the Open circuit characteristics at rated speed of the machine. To Conduct Load Test on the separately excited DC Generator and determine the efficiency and Regulation.
7. Output waveform of the single phase bridge rectifier and to design a filter capacitor to improve the dc output.
8. Three phase controlled rectifier and output waveform of the rectifier with different firing angle.
9. The working principle of the Insulated-Gate Bipolar Transistor (IGBT) based three phase Inverter and study the gate trigger circuit, their timing sequence and the output waveform.

**Reference:** Laboratory Manual.

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

**Objectives:** The course familiarizes the basic data structures such as arrays, linked lists, stacks, queues, heaps, binary trees, and graphs. The students will be able to learn to develop generic data structure classes containing operations such as insertion, deletion, searching, and sorting of each data structure. In addition, the students also learn to design and analyze efficiency of similar kind of algorithms using Big-O notation.

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

1. Differentiate between basic data types and abstract data types.
2. Understand importance of data structures in programming.
3. Identify the applications of various types of data structures such as lists, stacks, queues, binary search trees, graphs, etc.
4. Analyze the strength and weakness of the different data structures (including array or linked list implementations).
5. Compare various sorting and searching algorithms.
6. Understand various file structures.

**Prerequisite:** None

**Contents:**

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

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

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

**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.

**Disjoint set data structure:** Make-set, Union and Find Operations.

**Trees:** Heaps, Binary search trees (BST), heights of BST BALANCED BSTs: Red Black trees, AVL Trees, 2,3,4-trees, B Trees.

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

**Text Book:**

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

**Reference Book:**

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

**Objectives:** The laboratory course is designed to develop skills to design, implement, and analyze linear and non-linear generic data structures. It improves the programming ability of the students to identify and implement the suitable data structure for the given computational problem.

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

1. Develop necessary programming skills required to solve the given problems.
2. Select an appropriate data structure for solving a given computational problem.

3. Implement generic class for a given data structure using any an appropriate programming language.
4. Compare and analyze programs with same problems but different logics. For example, compare various sorting algorithms over different input sets.
5. Work and contribute as a team member among their peers.

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

**Prerequisite:** None

**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 findmax) on Binary Search trees.
9. Design and Implement basic operations of various balanced BSTs, e.g., AVL tree, 2-3 Trees.
10. Implementation of Breadth First Search Techniques.
11. Implementation of Depth First Search Techniques.

**Reference:** Laboratory instructions and handouts.

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

**Objectives:** The objectives of the course is to:

1. Acquaint students the interdisciplinary nature of the course.
2. Enlighten them about the history and philosophy of Science, Technology and Society in India.
3. Help students socialize and develop critical thinking skills.
4. Familiarize students with the changing nature cum interrelation of science, technology and society.

**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.

**Prerequisite:** None

### Contents

#### Introduction:

*Introduction to STS as a field of study and research in the twentieth century:* STS as Quest for Knowledge, Nature of Science, Distinct Characteristics of Science, Science in Twentieth Century, Science and Technology in Societal Context, and Key Concepts and Theorists in Cyber-culture Studies.

*Philosophical, Historical and Sociological Approaches to STS:* Philosophy of Science and its Interconnection with Technology and Society, Social Role and Function of Science, Science and Western Civilization, Social Theorists, Literature and Science, Culture of STS, Scientific/Technological and Sociological Literacy, Society and Culture, Sociology of Society to STS, Social world, Science and Technology a Social Institution, Technoscience and Globalizations, Life After STS.

*The growth and identity of Modern Science and Technology in India:* History, key themes, Turn to technology and Beyond, Professional associations, Journals, Deliberative Democracy, Modernism and Postmodernism, Pace of Innovation, Privileged Positions of Business and Science, STS Social Construction, and Technoscience.

#### Science Communication: Institutions, Ideologies, Practices

*The diversity of science communication in colonial India Communication in India:* Historical Perspective, Current State of Affair, Science Policy and Science Communication, Modes and Means of Science Communication in India, Role of Various Organizations, Role of Public Funded Institutions, and Challenges.

*Science Communication and the Nehruvian Agenda:* Jawaharlal Nehru and Mahatma Gandhi, Concept of Scientific Temper, Nature of Nehruvian Science, Features of Nehru's Developmental Strategy, Nehru and STS, Nehruvian Science and Postcolonialism, Scientific Policy Resolution, and Problems Plaguing Practice of Science in India.

*The ideology and image of developmental science:* Role of Technical Education, Engineering Profession, Wealth and Engineering, Engineering Institutions, Quality Assurance, Engineering Education, and Role of Various Agencies in Technical Education.

*The agenda of People's Science:* Educational Technologies and Pedagogy, History and Pedagogical Discussions, Computers and Learning, Constructivist Theory, Educational Use of Computers, Technology and Teaching, Affordances for Learning and Teaching, and Online Learning Environments.

*Liberalization and the commoditization of science and technology:* Impact of Science on Society, Globalization, Lifelong learning and the Learning Society, Privatization, Learning Science Outside the Classroom, ICT as Secondary Science, Innovation in Instructional Technology, Ethics, and Cybercrime.

#### Text Books:

1. *Science, Technology and Medicine in Colonial India* – David Arnold (Cambridge, 2004).
2. *Western Science in Modern India, Metropolitan Methods, Colonial Practices* – Pratik Chakrabarti, (Permanent Black, 2004).
3. *Cyberculture: The Key Concepts* by David Bell.
4. *Cyberculture Theorists* by David Bell.
5. *Science, Technology and Society: A Sociological Approach* by Wenda K. Bauchspies.
6. *Philosophy of Science: A Contemporary Introduction* by Allen Rosenberg.

#### Reference Books:

1. *A Concise History of Science in India* – D. M. Bose, S. N. Sen, and B.V. Subarayappa (Universities Press, 2009).
2. *Science and Society in Twentieth Century* by Wendy R. Sherman.
3. *Nature of Science in Science Education* by William F. McComas

## Course Structure: Second Year

### Semester-III

#### Common to CSE and IT Branch

Course Code	Course Name	L	T	P	C
SC201	Environmental Science	2	0	0	2
MA201	Probability and Statistics	3	1	0	4
HS201	Technical Writing	1	1	2	3
CS201	Object Oriented Design & Programming	3	0	0	3
CS261	Object Oriented Design & Programming Lab	0	0	3	2
CS203	Design and Analysis of Algorithms	3	0	0	3
CS263	Design and Analysis of Algorithms Lab	0	0	3	2
EC201	Digital Logic Design	3	0	0	3
EC261	Digital Logic Design Lab	0	0	2	1
<b>Total</b>		<b>15</b>	<b>2</b>	<b>10</b>	<b>23</b>

### Semester-IV

#### [A] CSE Branch

Course Code	Course Name	L	T	P	C
MA202	Numerical Techniques	0	1	2	2
HS202	Economics	3	0	0	3
CS202	System Software	3	0	0	3
CS204	Database Management System	3	0	0	3
CS262	Database Management System Lab	0	0	3	2
CS206	Operating Systems	3	0	0	3
CS266	Operating Systems Lab	0	0	3	2
CS208	Computer Organization and Architecture	3	0	0	3
CS268	Computer Organization and Architecture Lab	0	0	2	1
<b>Total</b>		<b>15</b>	<b>1</b>	<b>10</b>	<b>22</b>

#### [B] IT Branch

Course Code	Course Name	L	T	P	C
HS202	Economics	3	0	0	3
IT202	Web Technology	3	0	0	3
IT262	Web Technology Lab	0	0	2	1
CS204	Database Management System	3	0	0	3
CS262	Database Management System Lab	0	0	3	2
CS206	Operating Systems	3	0	0	3
CS266	Operating Systems Lab	0	0	3	2
CS208	Computer Organization and Architecture	3	0	0	3
CS268	Computer Organization and Architecture Lab	0	0	2	1
<b>Total</b>		<b>15</b>	<b>0</b>	<b>10</b>	<b>21</b>

**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	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:** MA101.

**Contents:**

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

**Random Variables:** 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.

**Special Distributions:** Discrete uniform, binomial, geometric, negative binomial, hypergeometric, Poisson, continuous uniform, exponential, gamma, beta, normal, lognormal, inverse Gaussian, Cauchy, double exponential distributions, reliability and hazard rate, reliability of series and parallel systems.

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

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

**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.

**Estimation:** 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, confidence intervals for proportions.

**Testing of Hypotheses:** Null and alternative hypotheses, the critical and acceptance regions, two types of error, power of the test, the most powerful test and Neyman-Pearson Fundamental Lemma, tests for one sample and two sample problems for normal populations, tests for proportions, Chi square goodness of fit test and its applications.

**Text Book:**

1. *Probability and Statistics for Computer Scientists*, by Michael Baron, CRC Press, second edition (2013).

**Reference Books:**

1. Athanasios Papoulis and S. Unnikrishna Pillai, ‘*Probability, Random Variables, and Stochastic Processes*’, Fourth Edition, McGraw-Hill, Europe, 2002.
2. Alberto Leon-Garcia, ‘*Probability, Statistics, and Random Processes for Electrical Engineering*’, Third Edition, Pearson, 2008.

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

**Objectives:** The objectives of this course include understanding the concepts, terms and tools of technical writing; and applying them on various forms of

representation like technical reports, projects, research papers, dissertation and thesis. Students will read, analyze, and interpret material from technical fields, and will practice research and writing skills appropriate for technical topics.

**Learning Outcomes:**

1. Students will learn to follow the steps of writing process, i.e., pre-writing, writing, rewriting, and editing, and apply them to technical and workplace writing tasks.
2. Students will be able to prepare technical documents including project reports, manuscript preparation for conferences and journals, and drafting reports for availing grants for technical projects.
3. Students will be exposed to various software tools (say, LaTeX) for preparing technical documentation and presentation.
4. Students will understand the basic components of definitions, descriptions, process explanations, and other common forms of technical writing.
5. Students will have an appreciation for some of the ideas, issues, and problems involved in writing about technology and in workplace writing.

**Prerequisite:** None

**Contents:**

Structure of sentences, paragraphs, and documents. using stress for emphasis, and sequencing topics to create forward flow, writing for the reader; Formats of technical documents; the experimental report; the technical report, the proposal; workshop on published documents; Discussion and workshop on term paper proposals; Graphics; emphasis without distortion; visual illusions; a minimalist approach to data representation; univariate and multivariate displays; Discussion and workshop on term papers; elements of oral presentations; oral presentations.

**Text Book:**

1. *The Elements of Style*, W. Strunk, E B White, New York: Macmillan, 1972.

**Reference Books:**

1. *The Mayfield Handbook of Technical and Scientific Writing*, L. Perelman, Mayfield Publishing Company, 1998.
2. *The Science of Scientific Writing*, G. D. Gopen, J. A. Swan, American Scientist, 78(6):550-558, Nov-Dec 1990.

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

**Objective:** 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:** IT101.

**Contents:**

**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.

**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.

**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.

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

**Object Oriented Design:** Requirements modeling, business modeling, component based development; Rational Unified Process (RUP), process overview, phases and iterations, static structure of the process, core workflows; UML history, building blocks of UML, structural modeling, behavioral modeling; Use Case Diagrams, Modeling Ordered Interactions: Sequence Diagrams; case studies.

**Text Books:**

1. *Introduction to object-oriented programming*, B. Timothy, Pearson, 2001.
2. *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development*, C. Larman, Prentice Hall, 2004.



**Reference Books:**

1. *Object Oriented Design and Patterns*, C. Horstmann, John Wiley & Sons, 2005.
2. *Unified Modeling Language User Guide*, G. Booch, J. Rumbaugh, I. Jacobson, Pearson Education, 2001.
3. *Object-Oriented Systems Analysis and Design using UML*, Bennett, McRobb, Farmer, McGraw-Hill, 2002.

Course Code	Course Name	L-T-P: C
CS261	Object Oriented Design & Programming Lab	0-0-3: 2

**Objective:** This course equip students with object-oriented programming skills required to build reusable, robust and maintainable softwares using Java.

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

**Prerequisite:** IT101

**Contents:**

Lab and take home assignments based on the course ‘Object Oriented Design and Programming’. Emphasis on following topics:

1. Eclipse (or NetBeans) IDE introduction.
2. Compiling & running programs on IDE.
3. Object oriented coding conventions.
4. Simple example of object-oriented design and message passing.
5. Problems on object based iteration.
6. Problems on object based arrays, matrices, and strings.
7. Design oriented problems on object polymorphism.
8. Design oriented problems on object inheritance & overriding.
9. Object-oriented designing of advanced data structures (linked list, trees, graphs, tables).
10. Problems on object based linked lists.
11. Problems on object based trees.
12. Problems on object based graphs.
13. Mini Projects.

**Text Books:**

1. *Big Java/Big C++*, Horstmann, Cay S., John Wiley & Sons
2. *Object-Oriented Analysis and Design with Applications*, Brooch, Grady, Addison Wesley.

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

**Objectives:** Algorithms are at the core of computer science, and this course intends to provide a rigorous introduction to fundamental techniques in the design and analysis of algorithms. The course will be divided into five major components namely,

1. Foundations,
2. Sorting and Order Statistics,
3. Advanced Design and Analysis Techniques,
4. Graph Algorithms and
5. Special Topics.

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

1. Apply existing algorithms and analyze their strengths and weaknesses.
2. Design, implement, and analyze new algorithms.

**Prerequisites:** MA102, IT101 and CS102.

**Contents**

**Introduction and asymptotic notations:** The role of algorithms in computing, Insertion sort, Analysing algorithms (Random-access machine (RAM) model), Designing algorithms, Asymptotic notations.

**Divide and Conquer Techniques:** Divide and conquer algorithms such as the maximum-subarray problem, Strassen’s algorithm for matrix multiplication, etc., Solving recurrences -- The substitution method, The recursion-tree method, The master method, Proof of the master theorem.

**Heapsort and Quicksort:** Heaps, Maintaining the heap property, Building a heap, the heapsort algorithm, Priority queues, Quicksort.

**Dynamic Programming:** Dynamic programming algorithms such as the rod cutting, matrix-chain multiplication, Longest common subsequence, etc., Elements of dynamic programming – Optimal Substructure, Overlapping sub-problems.

**Greedy Algorithms:** Greedy algorithms such as activity-selection problem, huffman codes, etc., Elements of the greedy strategy - Optimal Substructure, Greedy choice property.

**Graph Algorithms:** Representations of graphs, Depth First search, Breadth First Search, Topological sort, Minimum Spanning Trees - The algorithms of Kruskal and Prim, Shortest Paths - The Bellman-Ford algorithm, Dijkstra’s algorithm.

**NP-Completeness and Approximation Algorithms –** Introduction to NP-Completeness, Approximation algorithms such as the traveling-salesman problem, the subset-sum problem, etc.

**Text Book:**

1. Cormen *et al.*, *Introduction to Algorithms*, Third Edition, Publisher: The MIT Press, 2009.

**Reference Books:**

1. Goodrich and Tamassia, *Algorithm Design and Applications*, First Edition, Publisher: Wiley, 2014.
2. Sedgewick and Wayne, *Algorithms*, Forth Edition, Publisher: Addison-Wesley Professional, 2011.

Course Code	Course Name	L-T-P: C
CS263	Design and Analysis of Algorithms Lab	0-0-3: 2

**Objectives:** The objectives of the lab course is to let the students learn to design and implement the state-of-the-art algorithms, empirically analyze the performance (e.g., running time, etc.) of algorithms and to identify suitable algorithms for resource-constrained devices.

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

1. Implement algorithms using different algorithm design paradigms such as divide-and conquer paradigm, greedy paradigm, and dynamic-programming paradigm.
2. Empirically analyze the performance (e.g., running time, etc.) of different algorithms.
3. Comparatively evaluate the performance (e.g., running time, etc.) of different algorithms.
4. Implement the state-of-the-art graph algorithms. Empirically analyze the performance of graph algorithms.

**Prerequisites:** MA102, IT101, IT161, CS102 & CS162.

**Contents:**

Laboratory assignments should be given from the following topics (or any other state-of-the-art topic) to develop creativity and analytical abilities of students. The assignments should be in the form of hands-on experience of algorithms in one (or more) of the programming language(s).

1. Divide and conquer algorithms.
2. Sorting techniques such as Merge sort, Quick sort, etc
3. Heap sort and Priority Queues.
4. Greedy algorithms.
5. Graph representation techniques.
6. Graph search. (DFS, BFS, etc.)
7. Minimum spanning trees. (Prim, Kruskal, etc.)
8. Shortest paths algorithms. (Bellman-ford algorithm, Dijkstra’s algorithm)
9. Dynamic programming-Algorithms.

**References:** Laboratory instructions and handouts.

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.

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

1. Understand the fundamentals of digital logic design, Boolean algebra and number systems.
2. Understand the combinational logic circuits.
3. Understand the sequential logic circuits.
4. Understand the design and implementation of digital circuits using VHDL.

**Prerequisite:** EC100

**Contents:**

**Number Systems:** Representations, signed, 1’s complement, 2’s complement, saturation and overflow in fixed point arithmetic.

**Boolean Algebra:** Axioms and theorems, De Morgan’s law, universal gate, duality, expression manipulation using axioms and theorems.

**Combinational Logic:** Introduction to switching algebra, canonical forms, two-level simplification, Boolean cube, logic minimization using K-map method, QuineMcCluskey tabular method, minimization for product-of-sum form, minimization for sum-of-product form, multiplexers, demultiplexers, decoders, encoders, hazard free synthesis, Arithmetic circuits, adders, half adder, full adder, BCD adder, ripple carry adder, carry-look ahead adder, combinational multiplier.

**Sequential Logic:** Simple circuits with feedback, basic latches, clocks, R-S latch, master- slave latch, J-K flip flop, T flip-flop, D flip-flop, storage registers, shift register, ripple counter, synchronous counters, Finite State Machine (Moore/Mealy Machines), FSM with single/multiple inputs and single/multiple outputs, RAM, ROM, EPROM.

**Hardware Description Language:** Programming and simulation, structural specification, behavioral specification, dataflow modelling, test bench, testing using test vectors, testing using waveforms, design of basic blocks to build larger circuits, case studies, adder, ALU, counters, shift registers, register bank, FSM design example etc.

**Text Books:**

1. *Digital Design*, Morris Mano, Prentice Hall, 2002.

2. *Digital Fundamentals*, 10<sup>th</sup> 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 Practises*, 4<sup>th</sup> Ed, J F Wakerly, Prentice Hall, 2006.

9. Write VHDL programme for the following, analysis the output waveform and the hardware generated (a) Arithmetic logic unit (ALU), (b) Finite state machine (FSM).

**References:** Laboratory Manual.

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

**Objective:** To provide hands-on experience in digital circuits, which can be design and implemented by using standard integrated circuits (ICs). To investigate the operation of several digital combinational and sequential circuits. To provide the basic understanding of design and implementation of the digital circuits using VHDL

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

1. Minimize the Boolean algebra using the minimization methods (K-Map and Quine–McCluskey)
2. Realize the logics circuits using the universal gates.
3. Analyze and design a combinational circuit.
4. Realize the given function using the combinational circuit.
5. Design and develop sequential circuits.
6. Understand the design and implementation of the digital circuits using VHDL

**Prerequisite:** EC100

**List of Experiments:**

1. Verification and interpretation of truth tables for logic gates.
2. Realization of logic functions with the help of universal gates-NAND Gate & NOR Gate.
3. Construction of half adder, full adder and full subtractor using XOR and NAND gates and verification of its operation.
4. Construction of a SR and JK flip flop and verification of its operation, convert SR to JK, JK to D, SR to T, J-K to T flip flop.
5. Verification of truth table for 7 segment decoder/driver ICs.
6. Verification of truth table for 2:1, 4:1, 8:1 multiplexer/ Demultiplexer and implement 16:1 using 4:1 multiplexer.
7. Construction of 4 bit SISO, SIPO, PISO, PIPO shift registers and verification of their operation.
8. Construction and verification of operation of 4-bit ring counter.

**Semester-IV: Courses and Contents**

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

**Objective:** This course provides insights in to 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 algorithm for specific problem.
2. Apply algorithm with analysis of cost benefits.

**Prerequisite:** MA101.

**List of Experiments:**

1. **Solving system of linear algebraic equation:** Gauss Elimination Method, LU Decomposition, Iterative Methods, Singular Value Decomposition, Ill Conditioned System, Eigen Value Computation (Jacobi Method for Symmetric Matrices)
2. **Interpolation and Curve Fitting:** Polynomial interpolation, Extrapolation, Cubic Spline, Least Square fit.
3. **Finding roots of an Equation:** Incremental Search method, Bisection Method, Newton Raphson Method, Regula-Falsi method.
4. **Numerical Differentiation:** Finite Difference Approximation, Derivatives by Interpolation (Using Forward/ Backward/central difference formula)
5. **Numerical Integration:** Trapezoidal and Simpson's rules for integration.
6. **Solution of first order and second order ordinary differential equations:** Euler method, Euler modified method, Runge-Kutta methods, Milne PC method, Boundry Value Problem.

**Text Books:**

1. *Numerical Methods in Engineering with Python*, Jaan Kiusalaas, 2<sup>nd</sup> Ed. Cambridge University Press.
2. *Numerical Recipes in C*, W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, 3<sup>rd</sup> Ed., Cambridge University Press.

**Reference Books:**

1. *Elementary Numerical Analysis*, S. D. Conte and C. De Boor, Mc Graw Hill Publisher.
2. *Applied Numerical Analysis*, E. V. Krishnamurthy & S. K. Sen, East West Publication.
3. *Numerical Methods*, B. Ram, Pearson Education.

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

**Objectives:** To integrate the basic concepts of economics with the tools of mathematics and statistics in order to analyze and make optimal business decisions.

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

1. Understand the roles of the Organizations.
2. Analyze the demand and supply conditions and assess the position of a company.
3. Design competition strategies, including costing, pricing, product differentiation, and market environment according to the natures of products and the structures of the markets.
4. Analyze real-world business problems with a systematic theoretical framework.
5. Make optimal business decisions by integrating the concepts of economics, mathematics and statistics.

**Prerequisite:** MA101, MA201

**Contents:**

The Problems of Economic Organization; Demand and Supply; Price Determination; Elasticity of Demand and Supply; Theory of Production; Production function; Law of diminishing returns; Analysis of Cost; Fixed and variable costs; Marginal cost; Market Structure and Various Types of Markets; Perfectly Competitive Market; Monopolistic Markets; Aggregate Demand and Aggregate Supply; Determination Of National Income and criticisms; Consumption, Saving and Investment; Business Cycle and remedies; International Trade; Balance of Payment; Case for and against free trade; Economics of banking; Interest rates and demand for money; Role of Central Bank; Inflation: measurement, causes and index numbers.

**Text Books:**

1. *Economics*, P. Samuelson & Nordhaus, Tata-McGraw Hill.
2. *Indian Economy*, Ruddar Datt & Sundaram, S. Chand & Co.

Course Code	Course Name	L-T-P: C
CS202	System Software	3-0-0: 3

**Objective:** 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.

- Analyze different types of software processors viz. assemblers, compilers, loaders.
- Able to differentiate between top down and bottom up parsing and understand syntax directed translation techniques.

**Prerequisites:** CS102.

**Contents:**

**Introduction:** Overview and history, Language Processors, Introduction to CISC and RISC machine architecture.

**Assembler:** Basic Assembler Functions, Machine Dependent Features, Machine Independent Features, One pass and Multi pass Assembler.

**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.

**Compilers:** Aspects of Compilation, Compiler Features, Memory Allocation, Grammar, Parsing Techniques, Compiler Design Options, Intermediate Code Generation and Optimization Techniques.

**Software Tools:** Text Editors, Debuggers, Data Base Management System, User Interfaces.

**Text Books:**

- System Software – An introduction to System Programming*, Leland L. Beck, 3<sup>rd</sup> Edition, Pearson Education.
- Systems Programming and Operating Systems*, D. M. Dhamdhare, Tata McGraw Hill Publication.

**Reference Books:**

- System Programming*, John Donovan, McGraw Hill Publication.
- System Software*, Santanu Chattopadhyay, Prentice Hall India Publication.

Course Code	Course Name	L-T-P: C
IT202	Web Technology	3-0-0: 3

**Objectives:** Students will learn to use 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 application domain.

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

- Know the history of the WWW and associated technologies. The client-server architecture and

associated communication protocols such as HTTP, HTTPS, etc.

- Formats and languages used in modern WWW: HTML, XHTML, CSS, XML, XSLT, Javascript, DOM, etc.
- Understand basics of client-server web application.
- Know the usage and different ways to connect database with a web application and relevant data access operations.

**Prerequisites:** CS201

**Contents:**

**Introduction Web Services:** Web services architecture; overview of web services; service oriented roles and architecture; architectural process; three tier web based architecture.

**XML:** Introduction to XML; XML fundamentals; well-formed XML documents; components of XML document; XML tools; XML style sheets; XSL; CSS; XML namespaces; EDI fact; message definition; segments; message structure and electronic enveloping.

**Java Web Services Architecture:** J2EE and web services-Introduction to JSP and java servlets; servlets; overview of Java server pages.

**Active Server Pages:** HTML and VBScript fundamentals; ASP concepts, using request, response, application, session, server objects; cookies.

**.Net Framework:** Overview of .NET framework; building blocks of .NET platform; role of .NET class libraries; understanding CTS, CLR, CLS; deploying .NET; building C# applications.

**Text Book:**

- Web Application Architecture: Principles, Protocols, & Practices*, L. Shklar, R. Rosen.

**Reference Books:**

- Web Technologies: A Computer Science Perspective*, Jeffrey Jackson.
- Web Technology: Theory and Practice*, M. Srinivasan.
- Java Servlet Programming*, J. Hunter, W. Crawford, O'Reilly Publications, USA, 1998.

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:

1. Design and implement dynamic websites using appropriate and modern languages discussed.
2. Create a client-server web application using appropriate tools.
3. Connect a webpage to a database and perform relevant data access operations.
4. 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.

**Reference:** Laboratory instructions and handouts.

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:

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

**Prerequisites:** CS102

**Contents:**

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

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

**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.

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

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

**Object and Object Relational Databases:** Object database concepts, the ODMG standard for object databases, object-relational systems, and SQL.

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

**Text Book:**

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

**Reference Books:**

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

Course Code	Course Name	L-T-P: C
CS262	Database Management System Lab	0-0-3: 2

**Objective:** This course aims to develop an in-house project using the fundamentals of Database Management System design process discussed in the course "Database Management System".

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

1. Design and implement a database schema.
2. Use normalization techniques in a real-time DBMS application.

3. Formulate SQL commands such as create, insert, update, delete, etc. to a relational DBMS.

**Prerequisites:** CS102

**Contents:**

Lab and take home assignments based on the course “DBMS”. Emphasis on following topics:

1. **ER Modeling Tool (ERWin):** Introduction to ERWin; Adding Entity types & relations; Forward generation.
2. **Abstract Query Language Interpreter (JCup & JFlex):** Relational Algebra (syntax, RA interpreter); Domain Relational Calculus (syntax, DRC interpreter); Datalog (syntax, Datalog interpreter).
3. **Relational Database Management System (Oracle):** SQL\* Plus Utility; SQL\* Loader Utility; Programming with Oracle using JDBC API.
4. **Relational Database Management System (MySQL):** MySQL Utility; Bulk loading of data; MySQL and PHP programming; Making an online Address Book.
5. **Database Design Toolkit (DBD):** Coding Relational Schemas & Functional Dependencies; Invoking SWI-Prolog Interpreter; DBD system predicates (xplus, finplus, fplus, implies, equiv, superkey, candkey, mincover).
6. **Object-Oriented Database Management System (db4o):** db4o Installation & Introduction; Simple database creation exercise; Database updates & deletes; Database Querying (queryByExample, Native Queries, SODA Queries); Company database application exercise; Web application exercise (client-server configuration).XML DATABASE: XML basics; Creating a company database in XML; XML Editor (EditiX); XPath; XQuery; FLWOR expressions; XML Schema

**Reference:** Laboratory instructions and handouts.

Course Code	Course Name	L-T-P: C
CS206	Operating Systems	3-0-0: 3

**Objective:** This course provides undergraduate students with knowledge about contemporary operating system design and its relationship between memory, processor, file interface and processes.

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

1. Describe process management and concepts of threading, multitasking, IPC.
2. Differentiation of various scheduling algorithms and identify the reasons of deadlock and their remedial measures in an operating system.

3. Describe various memory management techniques, file system interfacing and disk scheduling algorithms.
4. Conceptualize the components involved in designing a contemporary OS.

**Prerequisites:** MA102 and IT101.

**Contents:**

**Introduction:** Overview and history, multi-programming, functions of an OS, device drivers, I/O interrupts, and system call interface.

**Process Management:** Process Abstraction, Process States, Implementing Processes (PCB), Threads, Classical Synchronization Problems; Synchronization Primitives; Semaphores; Monitors; Deadlocks, Deadlock Avoidance. CPU Scheduling, Real Time Scheduling.

**Memory Management:** Segmentation, Demand Paging, Hardware Support, Page Fault Handling; Page Replacement Algorithms; Shared Memory.

**Files Systems:** Disks Structure, Disk Scheduling, Disk Management, RAID Structure, File Organization and Mounting, File descriptor, Directory Structure and Implementation, I/O Systems.

**Text Books:**

1. *Operating Systems: internals and design principles*, William Stallings, Pearson Education.
2. *Modern Operating Systems*, Andrew Tanenbaum, Prentice Hall Publication.

**Reference Books:**

1. *Operating System Concepts*, A. Silberschatz, P. Galvin, & G. Gagne, John Wiley & Sons.
2. *Operating System – A Concept based Approach*, D. M. Dhamdhare, McGraw Hill Publication.

Course Code	Course Name	L-T-P: C
CS266	Operating Systems Lab	0-0-3: 2

**Objective:** This lab course provides practical implementation of various architectural algorithms used contemporary operating system. It includes analysis of design related issues using simulations.

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

1. Program using shell scripting.
2. Perform multithread programming on single core architecture using libraries.
3. Implement of various process scheduling algorithms.
4. Implement Deadlock avoidance method and analysis on dynamic system behavior.
5. Implement Paging Concept of Operating System.

**Prerequisites:** MA102 and IT101.

**List of Experiments:**

1. Review of Pointers and File Handling using C.
2. Understanding of Shell Scripting.
3. Understanding of System Calls.
4. Implementation of Producer-Consumer Problem using Stack & Queue.
5. Multithreaded Programming.
6. Implementation of basic CPU Scheduling algorithms like FCFS, SJF, Round Robin.
7. Implementation of Banker's Algorithm.
8. Implementation of basic Page Replacement Algorithm like LRU, Optimal Page Algorithm.

**Text Book:**

1. *Operating System Concepts*, A. Silberschatz, P. Galvin, & G. Gagne, John Wiley & Sons.

**Reference Books:**

1. *Operating Systems: Internals and Design Principles*, William Stallings, Pearson Education.
2. *Modern Operating Systems*, Andrew Tanenbaum, Prentice Hall Publication.
3. *Operating System – A Concept based Approach*, D. M. Dhamdhare, McGraw Hill Publication.

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

**Objectives:** To study the basic organization and architecture of digital computers (CPU, memory, I/O, software, pipelining and parallelism). Discussions will include digital logic, microprogramming and performance enhancement of processor. 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. To familiarize the students with the computer resources as designer and user point of view.
2. To provide the hands on experience of software and hardware.
3. To familiarize the students with the processor interaction with other hardware and time management.
4. To familiarize the students with pipeline architecture and instruction level parallelism.
5. To familiarize the students with dynamic instruction scheduling and thread level parallelism.
6. To familiarize the students with memory system and its interaction with processor.

**Prerequisite:** EC201

**Contents:**

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

**Data representation and arithmetic:** Overview of integer data, fixed, floating point systems, representation of non-numeric data (characters, strings, records, and arrays), integer addition and subtraction, ripple carry adder, carry look-ahead adder, etc. multiplication - shift-and-add, Booth multiplier, division, floating point arithmetic.

**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, super scaler architecture, overview of thread level parallelism.

**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. *Introduction to Computing Systems: From Bits and Bytes to C and Beyond*, 2th Ed, Yale Patt and Sanjay Patel, Tata McGraw-Hill, 2001.
2. *Computer Architecture: A Quantitative Approach*, 5th Edition By John L Hennessy & David A Patterson.

**Reference Books:**

1. *Computer Organization and Design: The hardware/software Interface* (ARM edition) by John L Hennessy & David A Patterson.
2. *Computer Organization and Architecture*, 8th Ed, Stallings, Pearson, 2010.
3. B. Parhami, *Computer Architecture: From Microprocessors to Supercomputers*, Oxford Univ. Press, New York, 2005.

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

**Objective:** To provide hands-on experience using the software and hardware to understand the performance improvement of the processor.

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



1. Understand the instruction set architecture (ISA).
2. Understand the design and simulations of ALU and data path architecture.
3. Understand the computer performance measurements.
4. Understand the pipeline design and execution of instructions with minimum hazards.
5. Understand the instruction level parallelism and dynamic instruction scheduling.
6. Understand the performance measurements using Tomasulo and scoreboard algorithm.

**Prerequisite:** IT101, EC201

**List of 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 Score Board algorithm.
10. To understand the implementation of instruction level parallelism using Tomasulo algorithm.

**Reference:** Laboratory instructions and handouts.

---

## Course Structure: Third Year

### Semester-V

#### [A] CSE Branch

Course Code	Course Name	L	T	P	C
CS301	Computer Networks	3	0	0	3
CS361	Computer Networks Lab	0	0	2	1
CS303	Software Engineering	3	0	0	3
CS363	Software Engineering Lab	0	0	3	2
CS305	Formal Language and Automata Theory	3	0	2	4
-	Program Elective (PE1)*	3	0	2	4
-	Program Elective (PE2)*	3	0	0	3
CS391	Design Project	0	0	0	3
<b>Total</b>		<b>15</b>	<b>0</b>	<b>9</b>	<b>23</b>

\*List of Program Electives is provided in Appendix-I

#### [B] IT Branch

Course Code	Course Name	L	T	P	C
CS301	Computer Networks	3	0	0	3
CS361	Computer Networks Lab	0	0	2	1
CS303	Software Engineering	3	0	0	3
CS363	Software Engineering Lab	0	0	3	2
IT301	Information Security	3	0	0	3
IT361	Information Security Lab	0	0	2	1
-	Program Elective (PE1)*	3	0	2	4
-	Program Elective (PE2)*	3	0	0	3
IT391	Design Project	0	0	0	3
<b>Total</b>		<b>15</b>	<b>0</b>	<b>9</b>	<b>23</b>

\*List of Program Electives is provided in Appendix-I

### Semester-VI

#### [A] CSE Branch

Course Code	Course Name	L	T	P	C
CS302	Artificial Intelligence	3	0	0	3
CS362	Artificial Intelligence Lab	0	0	2	1
CS304	Introduction to Cryptography and Network Security	3	0	0	3
CS364	Introduction to Cryptography and Network Security Lab	0	0	2	1
-	Program Elective (PE3)*	3	0	2	4
-	Program Elective (PE4)*	0	1	2	2
-	Elective from other Branch of Engineering (EO1)#	3	0	0	3
-	Open Elective (OE1)\$	3	0	0	3
<b>Total</b>		<b>15</b>	<b>1</b>	<b>8</b>	<b>20</b>

\*List of Program Electives is provided in Appendix-I.

#List of Electives from other Branch of Engineering is provided in Appendix-II.

\$List of Open Electives is provided in Appendix-III.

#### [B] IT Branch

Course Code	Course Name	L	T	P	C
IT302	Software Project Management	3	0	0	3
IT362	Software Project Management Lab	0	0	3	2
IT304	Management Information System	3	0	0	3
-	Program Elective (PE3)*	3	0	2	4
-	Program Elective (PE4)*	0	1	2	2
-	Elective from other Branch of Engineering (EO1)#	3	0	0	3
-	Open Elective (OE1)\$	3	0	0	3
<b>Total</b>		<b>15</b>	<b>1</b>	<b>7</b>	<b>20</b>

\*List of Program Electives is provided in Appendix-I.

#List of Electives from other Branch of Engineering is provided in Appendix-II.

\$List of Open Electives is provided in Appendix-III.

**Semester-V: Courses and Contents**

Course Code	Course Name	L-T-P: C
CS301	Computer Networks	3-0-0: 3

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

1. Relate the TCP/IP layered model with real-life data communication.
2. Analyze the requirements for an organizational network layout and give the most appropriate networking architecture and technologies suited.
3. Have a working knowledge of connection-less and connection-oriented protocols.

**Prerequisite:** None

**Contents:**

**Introduction:** 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).

**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.

**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.

**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.

**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

**Reference Book:**

1. *TCP/IP Illustrated Volume 1: The Protocols*, Second edition, K. R. Fall and W. Richard Stevens, publisher: Pearson.

Course Code	Course Name	L-T-P: C
CS361	Computer Networks Lab	0-0-2: 1

**Objective:** This course makes students aware of various cabling technologies used in different types of network. 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 network.

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

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.

Students are expected to create an internet-based application such as e-mail application, chat application, etc. In the third week, the students should finalize their application title and communicated it to their respective TA. The students can form groups with at most four students in each group. Mid-term evaluation of the project will be held in the 7th week and end-term evaluation in the last week.

**Prerequisite:** None

**List of Laboratory Tasks:**

1. Analyze the IITV 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	Software Engineering	3-0-0: 3

**Objectives:** The following are the objectives of the course:

1. Provides an understanding of the foundations of software engineering.
2. Provides a foundation for students who want to develop their career in the broad field of computing and specifically in the areas of Information and Communication technology.
3. Apply key engineering principles and mathematical models to application development projects.
4. Emphasize the complete lifecycle of the software development process and design, develop, test, and deploy software using rigorous software engineering practices.
5. Develop the skills and abilities of applying the fundamental concepts of computing in industrial, business and other problems, in order to produce software solutions.
6. Introduce the role of software tools in the process of software development.

This course has been designed to provide the students with the opportunity to apply the software engineering principles learned in the course to a project, that is also a part of this course. Students will work on a significant software development project that may include any or all activities associated with creating a software solution to a client/customer problem. They would be taken through all the software engineering activities that are typically experienced from the initiation to the completion of a software development project. Special emphasis has been placed on defining the client/customer problem and determining requirements by either working with real clients on real world problems. Teams are encouraged to work autonomously following good software engineering practices, with guidance in the form of lectures and tutorials, from the course instructor and

teaching assistants. In addition, issue based assistance is provided as and when required or as and when the same is sought by the team members.

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

1. Enumerate and define the phases in the software development process.
2. Describe the activities performed in each of the phases and how each phase relates to the others.
3. Develop a coherent set of software requirements for a particular application.
4. Convert a set of requirements into a set of specifications that can be validated.
5. Apply any of several design methodologies to the design of a software work product.
6. Develop and implement a test plan that will adequately exercise a software work product with the purpose of discovering defects.
7. Enumerate and define the steps in the post-implementation phases.
8. Describe the activities associated with corrective, adaptive, and perfective maintenance.
9. Describe the activities associated with the configuration management process and relate its importance during software development and maintenance.
10. Perform an impact analysis for a change request as it applies to a software work product.
11. Perform all software engineering tasks associated with developing a software system or product requiring a team of software engineers.
12. At the end of the project, assess a software development effort to determine the appropriate principles and practices that will maximize the probabilities for success.

**Prerequisite:** CS201.

**Contents:**

**Introduction to Software Engineering**

**Software life cycle models**

**Software Project Phases**

1. Pre-development phase-Feasibility & Proposal.
2. Development phase: Requirements analysis & specification, Design, Coding & Unit Testing, Testing.
3. Post-development phase: Maintenance

**Software Project Management**

1. Software Project Estimation
2. Software Project Scheduling
3. Risk Management
4. Configuration Management
5. Software Reliability and Quality Assurance

**Computer Aided Software Engineering (CASE)**

**Agile Methodologies**

**Additional Expectations:** The student should be able to:

1. Analyze a software development project and determine the most appropriate software engineering principles and practices for the given situation.
2. Evaluate the effectiveness of a given set of software engineering practices and make recommendations for changes that can improve the software development project.
3. Analyze a software development project to determine missing or inappropriate software engineering practices.
4. Assess the quality of software engineering processes, practices, products, and artifacts associated with a software engineering development effort.
5. Demonstrate interpersonal and team skills that support maximizing their team's effectiveness.

**Special Expectations:** The student should be able to:

1. Work collaboratively and cooperatively with others as a team that produces the required software engineering work products.
2. Create and deliver a quality presentation (individually and as part of a team presentation) related to selected aspects of software engineering processes, practices and work products associated with a software engineering project.

**Text Books:**

1. Roger S Pressman, “*Software Engineering – A practitioner’s Approach*”, McGraw Hill Higher Education, 5th, 6th and 7th Edition, 2010.
2. Pankaj Jalote, “*An Integrated Approach to Software Engineering*”, Narosa Publication, 2001.

**Reference Books:**

1. Eric Braude (Ed.), “*Software Engineering – Selected Readings*”, IEEE, 2000.
2. Ivor Jacobson, “*Object Oriented Software Engineering: A Use Case Driven Approach*”, Pearson Education Asia, 2001.
3. Carlo Ghezzi, “*Fundamentals of Software Engineering*”. Prentice Hall of India, 2001.
4. Ian Sommerville, “*Software Engineering*”, 8th Edition, Addison-Wesley, 2006.
5. Richard Fairley, “*Software Engineering Concepts*”, Tata McGraw-Hill, 2001.
6. Ali Behforooz, “*Software Engineering Fundamentals*”, Oxford University Press, 1996.
7. Richard Thayer (ed.), “*Software Engineering Project Management*”, IEEE Computer Society, 1997.
8. Rajib Mall, “*Fundamentals of Software Engineering*”, Prentice Hall of India, 2003.

Course Code	Course Name	L-T-P: C
CS363	Software Engineering Lab	0-0-3: 2

**Objectives:** To learn skills to work effectively as a team member and/or leader in a professional environment, typically a software industry.

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

1. Identify, formulate, and solve complex engineering problems by applying principles of computer engineering, science, and mathematics.

**Prerequisites:** IT101 and IT161.

**Laboratory Assignments:**

1. Practical implementation of various aspects of software design life cycle.
2. Management of software engineering design activities.
3. Work on various projects in team/group.
4. Study and hands-on various CASE tools that are used by IT Industries.

**Reference:** Laboratory instructions and handouts.

Course Code	Course Name	L-T-P: C
CS305	Formal Languages and Automata Theory	3-0-2: 4

**Objective:** 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:**

**Mathematical Preliminaries:** Review of Set theory, Functions and Relations, Graphs, Proof Techniques.

**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.

**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.

**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
IT301	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 thorough tools/techniques such as firewalls, PGP, etc.

**Prerequisites:** MA102 and IT101.

**Contents:**

**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).

**Asymmetric-key Cryptography:** Applications of asymmetric-key cryptosystem, RSA cryptosystem, Diffie-Hellman key exchange protocol, Digital signature.

**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.

**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).

**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
IT361	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.

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.

Course Code	Course Name	L-T-P: C
CS391/ IT391	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

**Semester-VI: Courses and Contents**

Course Code	Course Name	L-T-P: C
CS302	Artificial Intelligence	3-0-0: 3

**Objective:** The use of computer to solve complex problems is the fundamental theme of this course. The notion of intelligence being captured by problem solving ability reflects throughout 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.

**Learning Outcomes:**

1. Student 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 design expert systems and be able to decouple the domain specific knowledge and decision/ inference engine.
4. Student will be able to program computer to solve logical inference problems with uncertainty.

**Prerequisites:** MA102, IT101 and CS102.

**Contents:**

**Introduction and History:** The state of art; intelligent agents; structure; environment.

**Problem Solving and Game Playing:** 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.

**Knowledge, Logic and Reasoning:** Propositional logic; predicate logic; rules; forward and backward chaining; strong and weak slot fillers. The meaning of knowledge, production, semantic nets, schemata, frames; propositional logic; The first Order Predicate Logic; The Universal Quantifier; The Existential Quantifier.

**Planning:** Goal stack planning; non-linear planning; hierarchical planning; reactive systems.

**Reasoning Under Uncertainty:** Non-monotonic reasoning; logics; implementation; probability and Bayes theorem; certainty factors; Bayesian networks; Dempster Shafer theory.

**Introduction to and Design of Expert Systems:** What is an Expert System; advantages of Expert System; general concepts of Expert system, characteristics of Expert System; Expert System application and

domain. Introduction, rule-based system architecture, non-production system architecture, dealing with uncertainty; knowledge acquisition and validation; knowledge system building tools; selecting the appropriate problem; stages in the development of Expert system; errors in development stages; software engineering and expert systems.

**Text Books:**

1. *Artificial Intelligence – A Modern Approach*, S. Russell, P. Norvig, Pearson Education, New Delhi, 2002.
2. *A First Course in Artificial Intelligence*, Khemani D., Tata McGraw Hill, 2014.

**References:**

1. *Artificial Intelligence: A guide to intelligent systems*, Negnevitsky M., Pearson Education, 2005.
2. *The Quest for Artificial Intelligence: A history of ideas and achievements*, Nilsson N., Cambridge University Press, 2008.

Course Code	Course Name	L-T-P: C
CS362	Artificial Intelligence Lab	0-0-2: 1

**Objectives:** The use of computer to solve complex problems is the fundamental theme of this course. Identifying the difficult problems in terms of computation and interpreting softwares as intelligent agents is central theme of laboratory experiments. Modeling the problems in a way that can be solved using computer programs is a major objective of the laboratory course.

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

1. Apply the knowledge of data-structure and algorithms for implementing solution strategies efficiently.
2. Model a real world problem in a formal way with appropriate specifications, for example: search problems, multiplayer games and strategies, expert systems etc.
3. Select suitable search strategy for the given problem
4. Design expert systems and be able to decouple the domain specific knowledge and decision/ inference engine.
5. Program a general purpose computer to solve logical inference problems with uncertainty

**Prerequisites:** MA102, CS102 and CS203.

**Laboratory Assignments:**

1. State Space Search: Modeling and BFS.
2. State Space Search: Heuristic search, Best First Search, Hill Climbing.



3. Non-deterministic Search: Genetic Algorithm.
4. Path Finding: A\* on Visibility Graph.
5. Game Playing: Min-Max and Alpha-Beta Algorithm.
6. Sudoku: Constraint Satisfaction Problem.
7. K-means clustering.
8. Neural Network.
9. Decision Tree.
10. Fuzzy Expert System

**Reference:** Laboratory instructions and handouts.

Course Code	Course Name	L-T-P: C
CS304	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:** IT101, MA102, and CS301.

**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).

**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
CS364	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:** 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. Secure socket layers (SSL)/Transport layer security (TLS).
14. Secure shell (SSH).
15. Firewalls (e.g. iptables)

**Reference:** Laboratory instructions and handouts.

Course Code	Course Name	L-T-P: C
IT302	Software Project Management	3-0-0: 3

**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 Book:**

1. *Applied Software Project Management* -- A. Stellman and J. Greene: O'Reilly Media, 2005.

Course Code	Course Name	L-T-P: C
IT362	Software Project Management Lab	0-0-3: 2

**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
IT304	Management Information System	3-0-0: 3

**Objective:** 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 Structure: Fourth Year****Semester-VII****[A] CSE Branch**

Course Code	Course Name	L	T	P	C
CS401	Introduction to Distributed and Parallel Computing	3	0	0	3
CS461	Introduction to Distributed and Parallel Computing Lab	0	0	2	1
-	Program Elective (PE5)*	3	0	0	3
-	Program Elective (PE6)*	3	0	2	4
-	Elective from other Branch of Engineering (EO2)#	3	0	0	3
-	Open Elective (OE2)\$	3	0	0	3
CS491	Research/Industrial Internship	0	0	0	3
<b>Total</b>		<b>15</b>	<b>0</b>	<b>4</b>	<b>20</b>

\*List of Program Electives is provided in Appendix-I.

#List of Electives from other Branch of Engineering is provided in Appendix-II.

\$List of Open Electives is provided in Appendix-III.

**[B] IT Branch**

Course Code	Course Name	L	T	P	C
IT401	System Administration and Maintenance	2	0	0	2
IT461	System Administration and Maintenance Lab	0	0	3	2
-	Program Elective (PE5)*	3	0	0	3
-	Program Elective (PE6)*	3	0	2	4
-	Elective from other Branch of Engineering (EO2)#	3	0	0	3
-	Open Elective (OE2)\$	3	0	0	3
IT491	Research/Industrial Internship	0	0	0	3
<b>Total</b>		<b>14</b>	<b>0</b>	<b>5</b>	<b>20</b>

\*List of Program Electives is provided in Appendix-I.

#List of Electives from other branch of Engineering is provided in Appendix-II.

\$List of Open Electives is provided in Appendix-III.

**Semester-VIII:****Common to CSE and IT Branch**

Course Code	Course Name	L	T	P	C
CS490/ IT490	B. Tech. Project	0	0	36	18
<b>Total</b>		<b>0</b>	<b>0</b>	<b>36</b>	<b>18</b>

**Semester-VII: Courses and Contents**

Course Code	Course Name	L-T-P: C
CS401	Introduction to 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:**

Overview of C, UNIX and UNIX system calls.

**Introduction:** Definition of a distributed system, goals, hardware concepts, software concepts, the client-server model.

**Communication:** Layered Protocols, Remote Procedure Call, Remote Object Invocation, Message-Oriented and Stream-Oriented Communication.

**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.

**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.

**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
CS461	Introduction to Distributed and Parallel Computing Lab	0-0-2: 1

**Objectives:** The course aim 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:** IT101, 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
IT401	System Administration and Maintenance	2-0-0: 2

**Objectives:** The course introduces students to the roles and responsibilities of the system administrator in large organizations (e.g., data centers), and focuses on hands-on experience of IT infrastructure administration.

**Learning Outcomes:**

1. To list the roles and responsibilities of a system administrator in organizations.
2. To manage the resources (system, network, database, users, etc.) and security of an IT infrastructure.
3. To use the UNIX/Linux utilities and scripting languages effectively.
4. To design and manage policies governing IT infrastructure.

**Prerequisite:** CS301.

**Contents:**

**Introduction:** Roles and responsibility of the system administrator, Friction between UNIX and Linux, Linux distributions and examples, System-specific administration tools, Man pages and other authoritative documentations.

**Scripting and the Shell:** Shell basics - Command editing, Pipes and redirection, Variables and quoting, Common filter commands, bash scripting, Regular expressions, Python scripting, Scripting best practices.

**Bootstrapping the system:** Booting PCs, GRUB: The GR and Unified Boot loader, Booting to single-user mode, Working with startup scripts, init and its run levels, systemd, Rebooting and shutting down.

**Access Control and Users:** Traditional UNIX access control, Modern access control - Role-based access control, SELinux: security-enhanced Linux, PAM: Pluggable Authentication Modules, Kerberos - third-party cryptographic authentication, Access control lists, Real-world access control – su and sudo, Pseudo-users other than root, The /etc/passwd file, The /etc/shadow file, The /etc/group file, Adding and removing users, Managing users with system-specific tools.

**Controlling Processes:** Components of a process, The life cycle of a process, Signals, kill, Process states, nice and renice: influence scheduling priority, ps - monitor processes, Dynamic monitoring with top, prstat, and topas, The /proc filesystem, strace: trace system calls, Runaway processes.

**Periodic Processes:** cron - schedule commands, The format of crontab files, Crontab management, Linux and Vixie-cron extensions, Common uses for cron.

**The Filesystem:** Pathnames, Filesystem mounting and unmounting, File types, File attributes - The permission bits, The setuid and setgid bits, The sticky bit.

**Syslog and log files:** Syslog: the system event logger, Syslog architecture, Configuring syslogd, Syslog debugging, Alternatives to syslog, Linux kernel and boot-time logging, logrotate - manage log files, Condensing log files to useful information, Logging policies.

**Backups and restore:** Introduction and requirements for backups, Backup devices and media, Incremental

backups, dump and restore, archiving programs – tar and dd, Commercial backup products.

**Networking and Routing:** TCP/IP and its relationship to the Internet, Networking road map, Packet addressing, IP addresses, Routing, ARP, DHCP, Basic network configuration, Linux networking.

**Text Book:**

1. Evi Nemeth et al., *UNIX and Linux System Administration Handbook*, Fifth Edition, Publisher: Addison-Wesley Professional, 2017.

**Reference Books:**

1. Aelen Frisch, *Essential System Administration*, Third Edition, Publisher: O'Reilly Media, 2009.
2. Limoncelli et al., *The Practice of System and Network Administration*, Second Edition, Publisher: Addison-Wesley Professional, 2007.

Course Code	Course Name	L-T-P: C
IT461	System Administration and Maintenance Lab	0-0-3: 2

**Objectives:** The objective of the laboratory course is to provide the students a platform to learn to manage the IT infrastructure using different tools/techniques.

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

1. Automate the repetitive tasks such as installation using different scripting languages.
2. Empirically analyze the start-up scripts used during the booting and shutting down procedure.
3. Manage the users of the system, and implement access control policies.
4. Empirically analyze and control the performance of the system by analyzing and controlling different processes.
5. Empirically analyze the role of different file-systems, and manage the file-systems.
6. Generate/configure/debug Syslog and log files.
7. Effectively use the tools/techniques to perform the backups.
8. Effectively use, at least, one of the following (or any other state-of-the-art) DevOps tools: Docker, Kubernetes, Jenkins, Chef, Puppet, Ansible, etc.

**Prerequisites:** CS206 and CS266.

**Laboratory Contents:**

1. OS Installations, VM Installation, Configurations, Printing, etc.
2. To learn and effectively use shell scripts e.g., Bash, Perl, Python, to monitor and manage IT infrastructure.
3. To understand the bootstrapping and shutdown, and the related scripts involved during the processes.

4. To configure & debug the startup scripts.
5. Access control and user management.
6. To effectively manage (e.g., monitor, start, stop, schedule, etc.) the processes.
7. Mount/ unmounts/ configure/ access-control/ etc. the filesystem.
8. Network configuration and management.
9. To generate/ configure/ debug Syslog and log files.
10. To generate/ configure/ debug backups and restore.

Apart from the laboratory exercises, students are expected to learn and effectively use, at least, one of the following (or any other state-of-the-art) DevOps tools: Docker, Kubernetes, Jenkins, Chef, Puppet, Ansible, etc.

**Reference:** Laboratory instructions and handouts.

Course Code	Course Name	L-T-P: C
CS491/ IT491	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.

## Semester-VIII

Course Code	Course Name	L-T-P: C
CS490/ IT490	B. Tech. Project	0-0-36: 18

The B. Tech. Project (BTP) is an academic course. A student may pursue BTP

- 1. On-campus:** Under the supervision of the Institute's faculty members.
- 2. 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:

1. Software Development.
2. System Design and Simulation.
3. Hardware Development/Implementation.
4. Embedded System (Software & Hardware combined) Development/Implementation.
5. Theoretical Modeling.
6. Technical Study.
7. 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

1. A novel/new idea.
2. An extension of some previous research work.
3. An abstract problem.
4. 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:

1. Software Development.
2. System Design & Simulation.
3. Hardware Development/Implementation.

**Deliverables:** In general, the expected deliverables will include one or more of the followings

1. Product developed.
2. Code developed.

3. Research work in the form of Conference/Journal paper.
4. Software developed along with instructions and source code.
5. Comparative study (products/ methods/ designs).
6. Simulation study results.
7. 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:

1. Student must find an On-campus supervisor from IIIT Vadodara.
2. The student is expected to complete the project work assigned by the On-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 On-campus supervisor. This includes reporting, leave and working hours during the project tenure.
4. The student should report immediately to the On-campus supervisor in the event of exceptional circumstances like illness.
5. 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.
6. Student will be responsible for the plagiarism and copy-right issues.

### For Off-campus students:

1. Student will be allotted an On-campus mentor from IIIT Vadodara. Students are advised to meet prospective faculty member to act as mentor.
2. 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-I

### List of Program Electives for CSE / IT Branch

Program Verticals (PV)	Code
Artificial Intelligence and Data Analytics	PV1
Cyber Physical Systems	PV2
Security	PV3
Computational Science	PV4

PV	Course Name
<b>PE1-PE3 &amp; PE5-PE6 (3-0-0:3 / 3-0-2:4)</b>	
PV2	Human Computer Interaction
PV2	Enterprise Resource Planning
PV2	Software Project Management (CSE)
PV1, PV2	Advance DBMS
PV1, PV4	Speech Science
PV2	Embedded System
PV2	Wireless Sensor Networks
PV3	Web Application Security
PV2	Advance Computer Networks
PV1	Natural Language Processing
PV1	Data Analytics
PV2	Computer Graphics and Animation
PV1	Information Retrieval
PV1, PV2	Computer Vision
PV2	Compiler Design
PV4	Logic for Computer Science
PV4	Principles of Programming Language
PV2	Software Verification
PV2	Internet of Things
PV2, PV3	Cloud Computing Security
PV2, PV3	Cyber Security
PV3	Security Protocols
PV1, PV4	Data Compression
PV1, PV4	Bio-Informatics
PV1, PV4	Pattern Recognition
PV1	Modeling and Simulation
PV4	Advanced Computer Architecture
PV4	Approximation Algorithms
PV4	Scientific Computing
PV1	Graph Signal Processing
PV1, PV4	Mathematics and Big Data
PV4	Vector Space Projection
PV4	Numerical Optimization
PV2, PV4	Introduction to Distributed and Parallel Computing (IT)
PV1, PV2	Machine Learning
PV1, PV2	Deep Learning
PV1, PV2	Reinforcement Learning
PV3	Introduction to Cryptography
PV3	Block Chain
PV3	Number Theory and Cryptography

PV3	Post Quantum Cryptography
<b>PE4 (0-1-2:2)</b>	
PV4	Parallel Programming Lab
PV4	VHDL Lab
PV1, PV2	Robotics Lab
PV4	HPC Lab

## Appendix-II

### List of Electives form other Branch of Engineering (EO1 and EO2)

Code	Course Name
	Digital Image Processing
	Deep Learning
	Information Theory Coding
	Cognitive Science
	Soft Computing
	Advanced Image Processing
	Graph Theory
	Low Power Circuit Design
	Real-time System
	Nano Technology
	VLSI Design
	E-Commerce

## Appendix-III

### List of Open Electives (OE1 and OE2)

Code	Course Name
	Operation Research
	Network Flow Algorithms
	Professional Ethics
	Quantum Mechanics
	Quantum Models in Science and Engineering
	Introduction to Quantum Computation
	Game Theory
	Computational Physics