Curriculum-2018

Bachelor of Technology

in

Computer Science and Engineering

&

Information Technology

Indian Institute of Information Technology Vadodara
## Table of Contents

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

<table>
<thead>
<tr>
<th>Letter Grade</th>
<th>Points</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>AA</td>
<td>10</td>
<td>Outstanding</td>
</tr>
<tr>
<td>AB</td>
<td>9</td>
<td>Excellent</td>
</tr>
<tr>
<td>BB</td>
<td>8</td>
<td>Very Good</td>
</tr>
<tr>
<td>BC</td>
<td>7</td>
<td>Good</td>
</tr>
<tr>
<td>CC</td>
<td>6</td>
<td>Average</td>
</tr>
<tr>
<td>CD</td>
<td>5</td>
<td>Below Average</td>
</tr>
<tr>
<td>DD</td>
<td>4</td>
<td>Poor</td>
</tr>
<tr>
<td>F</td>
<td>0</td>
<td>Fail</td>
</tr>
<tr>
<td>P</td>
<td>-</td>
<td>Pass</td>
</tr>
<tr>
<td>I</td>
<td>-</td>
<td>Incomplete</td>
</tr>
</tbody>
</table>

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

‘I’ grade will be awarded in a course if the overall performance of the student is satisfactory in the course, but the student either misses the end-semester examination due to illness, accident/death in the family or obtains such an approval from the Dean-AP under exceptional circumstances. A student who misses the end-semester examination must apply for permission with reasoning and proof. An application not so supported will not be considered. Grade ‘I’ awarded for missing the end-semester examination will be converted into a performance grade (depending on the overall performance of the student in the course) after taking an examination equivalent to the end-semester examination of that particular course. An ‘I’ grade must be converted into a performance grade by the specified date in the academic calendar for the next semester, otherwise it will be converted into ‘F’ grade.

Performance Indices
[A] Semester Performance Index (SPI)
The performance of a student in a semester is indicated by the SPI. The SPI is the weighted average of the grade points obtained in all the courses registered by the student during the semester, calculated to two decimal places.

[B] Cumulative Performance Index (CPI)
An up-to-date assessment of the overall performance of a student from the time of entering the Institute is obtained by calculating the student’s CPI. The CPI is weighted average of the grade points obtained in all the courses registered for credit by the student after entering the Institute. The CPI is also calculated to two decimal places.

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

$$SPI = \frac{U_1G_1 + U_2G_2 + \ldots}{U_1 + U_2 + \ldots}$$

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

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

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:

<table>
<thead>
<tr>
<th>Category</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 hour Lecture (L) per week</td>
<td>1 Credit</td>
</tr>
<tr>
<td>1 hour Tutorial (T) per week</td>
<td>1 Credit</td>
</tr>
<tr>
<td>2 hours Laboratory (P) per week</td>
<td>1 Credit</td>
</tr>
<tr>
<td>3 hours Laboratory (P) per week</td>
<td>2 Credits</td>
</tr>
</tbody>
</table>

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

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

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

Distribution of Credits in 1st Year (in %)

<table>
<thead>
<tr>
<th>Code</th>
<th>Course Categories</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>MS</td>
<td>Mathematics &amp; Statistics</td>
<td>22</td>
</tr>
<tr>
<td>SC</td>
<td>Basic Sciences (Physics, Chemistry, Biology), Environmental Science</td>
<td>13</td>
</tr>
<tr>
<td>HS</td>
<td>Humanities, Social Science, Literature, Management, Soft-skills</td>
<td>10</td>
</tr>
<tr>
<td>PC</td>
<td>Core Engineering Courses (from the branch of study)</td>
<td>8</td>
</tr>
<tr>
<td>OC</td>
<td>Core Engineering Courses (from the other branches of Engineering)</td>
<td>16</td>
</tr>
<tr>
<td>PE</td>
<td>Program Electives (from the branch of study)</td>
<td>0</td>
</tr>
<tr>
<td>EO</td>
<td>Electives (from other branches of Engineering)</td>
<td>0</td>
</tr>
<tr>
<td>OE</td>
<td>Open Electives (Science / Humanities / Engineering: Not more than one from each)</td>
<td>0</td>
</tr>
<tr>
<td>PI</td>
<td>Projects, Internships (Research/Industrial)</td>
<td>0</td>
</tr>
<tr>
<td><strong>Total Credits</strong></td>
<td><strong>46</strong></td>
<td></td>
</tr>
</tbody>
</table>
Distribution of Credits in 2nd to 4th years of B. Tech. (CSE) Program:

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

*Total Credits for Degree is 172.

Distribution of Credits in 2nd to 4th years of B. Tech. (IT) Program:

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

*Total Credits for Degree is 171.

Distribution of Credits in 2nd to 4th years of B. Tech. (CSE) Program (in %)

Distribution of Credits in 2nd to 4th years of B. Tech. (IT) Program (in %)
Semester wise distribution of Credits:

### A. CSE Branch

<table>
<thead>
<tr>
<th>Cr.</th>
<th>SEM-I</th>
<th>SEM-II</th>
<th>SEM-III</th>
<th>SEM-IV</th>
<th>SEM-V</th>
<th>SEM-VI</th>
<th>SEM-VII</th>
<th>SEM-VIII</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>L</td>
<td>T</td>
<td>P</td>
<td>C</td>
<td>L</td>
<td>T</td>
<td>P</td>
<td>C</td>
</tr>
<tr>
<td>MS</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>4</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SC</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>5</td>
<td>5</td>
<td>2</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HS</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PC</td>
<td>61</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>5</td>
<td>3</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>5</td>
<td>14</td>
<td>17</td>
<td>13</td>
<td>8</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>PE</td>
<td>20</td>
<td>3</td>
<td>0</td>
<td>3</td>
<td>5</td>
<td>3</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>OC</td>
<td>17</td>
<td>3</td>
<td>0</td>
<td>3</td>
<td>5</td>
<td>3</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0</td>
<td>3</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>EO</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>OE</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PI</td>
<td>24</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>27</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Tot</td>
<td>172</td>
<td>14</td>
<td>3</td>
<td>10</td>
<td>23</td>
<td>45</td>
<td>7</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>2</td>
<td>10</td>
<td>15</td>
<td>10</td>
<td>22</td>
<td>15</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>23</td>
<td>3</td>
<td>0</td>
<td>9</td>
<td>23</td>
<td>21</td>
<td>15</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>1</td>
<td>8</td>
<td>20</td>
<td>15</td>
<td>0</td>
<td>4</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>0</td>
<td>0</td>
<td>36</td>
<td>18</td>
<td>0</td>
<td>36</td>
<td>18</td>
</tr>
</tbody>
</table>

Contact hrs. per week: 27 26 27 26 24 24 19 36
### B. IT Branch

<table>
<thead>
<tr>
<th>Cr.</th>
<th>SEM-I</th>
<th>SEM-II</th>
<th>SEM-III</th>
<th>SEM-IV</th>
<th>SEM-V</th>
<th>SEM-VI</th>
<th>SEM-VII</th>
<th>SEM-VIII</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>L</td>
<td>T</td>
<td>P</td>
<td>L</td>
<td>T</td>
<td>P</td>
<td>L</td>
<td>T</td>
</tr>
<tr>
<td><strong>MS</strong> 12</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td><strong>SC</strong> 12</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>5</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td><strong>HS</strong> 12</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td><strong>PC</strong> 62</td>
<td>3</td>
<td>0</td>
<td>3</td>
<td>5</td>
<td>3</td>
<td>0</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td><strong>PE</strong> 20</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td><strong>OC</strong> 17</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>6</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td><strong>EO</strong> 6</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td><strong>OE</strong> 6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>PI</strong> 24</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td><strong>Total</strong> 171</td>
<td>14</td>
<td>3</td>
<td>10</td>
<td>23</td>
<td>15</td>
<td>4</td>
<td>7</td>
<td>23</td>
</tr>
<tr>
<td>Contact hrs. per week</td>
<td>27</td>
<td>26</td>
<td>27</td>
<td>25</td>
<td>24</td>
<td>24</td>
<td>24</td>
<td>19</td>
</tr>
</tbody>
</table>
## Course Structure: First Year
(Common to all branch of studies)

### Semester-I

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

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

### Semester-II

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

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

### Semester-I: Courses and Contents

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>L-T-P: C</th>
</tr>
</thead>
<tbody>
<tr>
<td>MA101</td>
<td>Mathematics-I (Linear Algebra and Matrices)</td>
<td>3-1-0: 4</td>
</tr>
</tbody>
</table>

**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: $A\mathbf{x} = \mathbf{b}$

**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:**

**Reference Books:**

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>L-T-P: C</th>
</tr>
</thead>
<tbody>
<tr>
<td>PH100</td>
<td>Mechanics and Thermodynamics</td>
<td>3-1-0: 4</td>
</tr>
</tbody>
</table>

**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.


**Thermodynamics:**


**Text Books:**

**Reference Book:**

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>L-T-P: C</th>
</tr>
</thead>
<tbody>
<tr>
<td>PH160</td>
<td>Mechanics and Thermodynamics Lab</td>
<td>0-0-2: 1</td>
</tr>
</tbody>
</table>

**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.

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.

Reference: Laboratory Manual.

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>L-T-P: C</th>
</tr>
</thead>
<tbody>
<tr>
<td>TT101</td>
<td>Computer Programming and Problem Solving</td>
<td>3-0-0: 3</td>
</tr>
</tbody>
</table>

**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:**

**Reference Books:**
4. *Programming In ANSI C* by E. Balagurusamy.

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>L-T-P: C</th>
</tr>
</thead>
<tbody>
<tr>
<td>TT161</td>
<td>Computer Programming and Problem Solving Lab</td>
<td>0-0-3: 2</td>
</tr>
</tbody>
</table>

**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.

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>L-T-P: C</th>
</tr>
</thead>
<tbody>
<tr>
<td>EC100</td>
<td>Basic Electronic Circuits</td>
<td>3-1-0: 4</td>
</tr>
</tbody>
</table>

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.


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

Text Book:

Reference Books:
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, $A_{\text{mid}}$, 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.
10. Design an active low pass and high pass filter of different orders using Op-Amplifier.

Reference: Laboratory Manual.

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>L-T-P: C</th>
</tr>
</thead>
<tbody>
<tr>
<td>EC160</td>
<td>Basic Electronic Circuits</td>
<td>0-0-3: 2</td>
</tr>
</tbody>
</table>

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, $A_{\text{mid}}$, 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.
10. Design an active low pass and high pass filter of different orders using Op-Amplifier.

Reference: Laboratory Manual.

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>L-T-P: C</th>
</tr>
</thead>
<tbody>
<tr>
<td>HS101</td>
<td>Spoken and Written</td>
<td>2-0-2: 3</td>
</tr>
</tbody>
</table>

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:**

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

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
11. The Gift of the Magi – O’ Henry
12. The Monkey's Paw- W.W. Jacobs
Semester-II: Courses and Contents

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>L-T-P: C</th>
</tr>
</thead>
<tbody>
<tr>
<td>MA102</td>
<td>Mathematics-II (Discrete Mathematics)</td>
<td>3-1-0: 4</td>
</tr>
</tbody>
</table>

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.


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:

Reference Books:

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>L-T-P: C</th>
</tr>
</thead>
<tbody>
<tr>
<td>PH110</td>
<td>Waves and Electromagnetics</td>
<td>3-1-0: 4</td>
</tr>
</tbody>
</table>

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:


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.


**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:**

**Reference Books:**

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>L-T-P: C</th>
</tr>
</thead>
<tbody>
<tr>
<td>PH170</td>
<td>Waves and Electromagnetics Lab</td>
<td>0-0-2: 1</td>
</tr>
</tbody>
</table>

**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.

4. Measure Magnetic field of a paired coils in a Helmholtz arrangement with a Teslameter.
5. Verify Faraday Law and Induced e.m.f.
7. Microwave optic system to study properties of electromagnetic waves.

**Reference:** Laboratory Manual.

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>L-T-P: C</th>
</tr>
</thead>
<tbody>
<tr>
<td>EE100</td>
<td>Basic Electrical Engineering</td>
<td>3-1-0: 4</td>
</tr>
</tbody>
</table>

**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.

components and efficiency, starting and speed control of induction 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:**

**Reference Books:**

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>L-T-P: C</th>
</tr>
</thead>
<tbody>
<tr>
<td>EE160</td>
<td>Basic Electrical Engineering Lab</td>
<td>0-0-3: 2</td>
</tr>
</tbody>
</table>

**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.

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>L-T-P: C</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS102</td>
<td>Introduction to Data Structure</td>
<td>3-0-0: 3</td>
</tr>
</tbody>
</table>

**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:

Reference Book:

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.

Reference: Laboratory instructions and handouts.

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>L-T-P: C</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS162</td>
<td>Introduction to Data Structure Lab</td>
<td>0-1-2: 2</td>
</tr>
<tr>
<td>HS102</td>
<td>Science, Technology and Society</td>
<td>3-0-0: 3</td>
</tr>
</tbody>
</table>

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 inter-disciplinary 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:**

*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.


**Text Books:**

4. *Cyberculture Theorists* by David Bell.

**Reference Books:**

3. *Nature of Science in Science Education* by William F. McComas
## Course Structure: Second Year

### Semester-III
Common to CSE and IT Branch

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

### Semester-IV

#### [A] CSE Branch

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

#### [B] IT Branch

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>HS202</td>
<td>Economics</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>IT202</td>
<td>Web Technology</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>IT262</td>
<td>Web Technology Lab</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>CS204</td>
<td>Database Management System</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>CS262</td>
<td>Database Management System Lab</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>CS206</td>
<td>Operating Systems</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>CS266</td>
<td>Operating Systems Lab</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>CS208</td>
<td>Computer Organization and Architecture</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>CS268</td>
<td>Computer Organization and Architecture Lab</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td>15</td>
<td>0</td>
<td>10</td>
<td>21</td>
</tr>
</tbody>
</table>
Semester-III: Courses and Contents

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>L-T-P: C</th>
</tr>
</thead>
<tbody>
<tr>
<td>SC201</td>
<td>Environmental Science</td>
<td>2-0-0: 2</td>
</tr>
</tbody>
</table>

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)


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.


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:

Reference Books:

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>L-T-P: C</th>
</tr>
</thead>
<tbody>
<tr>
<td>MA201</td>
<td>Probability and Statistics</td>
<td>3-1-0: 4</td>
</tr>
</tbody>
</table>

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, estimates, 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 of normal populations, tests for proportions, Chi square goodness of fit test and its applications.

Text Book:

Reference Books:

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>L-T-P: C</th>
</tr>
</thead>
<tbody>
<tr>
<td>HS201</td>
<td>Technical Writing</td>
<td>1-1-2: 3</td>
</tr>
</tbody>
</table>

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 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:**


**Reference Books:**


---

**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, overloaded 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:**

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.
10. Problems on object based linked lists.
11. Problems on object based trees.
12. Problems on object based graphs.

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.

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>L-T-P: C</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS261</td>
<td>Object Oriented Design &amp; Programming Lab</td>
<td>0-0-3: 2</td>
</tr>
</tbody>
</table>

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

Reference Books:

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>L-T-P: C</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS263</td>
<td>Design and Analysis of Algorithms Lab</td>
<td>0-0-3: 2</td>
</tr>
</tbody>
</table>

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)

References: Laboratory instructions and handouts.

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>L-T-P: C</th>
</tr>
</thead>
<tbody>
<tr>
<td>EC201</td>
<td>Digital Logic Design</td>
<td>3-0-0: 3</td>
</tr>
</tbody>
</table>

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.


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:

**Reference Books:**

---

**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.

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.
Semester-IV: Courses and Contents

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>L-T-P: C</th>
</tr>
</thead>
<tbody>
<tr>
<td>MA202</td>
<td>Numerical Techniques</td>
<td>0-1:2: 2</td>
</tr>
</tbody>
</table>

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

**Learning Outcomes:** On successful completion of this course, students should be able to:
1. Implement numerical algorithm for specific problem.
2. Apply algorithm with analysis of cost benefits.

**Prerequisite:** MA101.

**List of Experiments:**

**Text Books:**

**Reference Books:**
3. Numerical Methods, B. Ram, Pearson Education.

**Course Code** | **Course Name** | **L-T-P: C**
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>HS202</td>
<td>Economics</td>
<td>3-0-0: 3</td>
</tr>
</tbody>
</table>

**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:**
2. Indian Economy, Ruddar Datt & Sundaram, S. Chand & Co.

**Course Code** | **Course Name** | **L-T-P: C**
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CS202</td>
<td>System Software</td>
<td>3-0-0: 3</td>
</tr>
</tbody>
</table>

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

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.


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

Text Books:

Reference Books:

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>L-T-P: C</th>
</tr>
</thead>
<tbody>
<tr>
<td>IT202</td>
<td>Web Technology</td>
<td>3-0-0: 3</td>
</tr>
</tbody>
</table>

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:
1. Know the history of the WWW and associated communication protocols such as HTTP, HTTPS, etc.
2. Formats and languages used in modern WWW: HTML, XHTML, CSS, XML, XSLT, Javascript, DOM, etc.
4. 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:

Reference Books:

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>L-T-P: C</th>
</tr>
</thead>
<tbody>
<tr>
<td>IT262</td>
<td>Web Technology Lab</td>
<td>0-0-2: 1</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>L-T-P: C</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS204</td>
<td>Database Management System</td>
<td>3-0-0: 3</td>
</tr>
</tbody>
</table>

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:

Reference Books:

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>L-T-P: C</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS262</td>
<td>Database Management System Lab</td>
<td>0-0-3: 2</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>L-T-P: C</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS206</td>
<td>Operating Systems</td>
<td>3-0-0: 3</td>
</tr>
</tbody>
</table>

**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.


**Text Books:**

**Reference Books:**

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>L-T-P: C</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS266</td>
<td>Operating Systems Lab</td>
<td>0-0-3: 2</td>
</tr>
</tbody>
</table>

**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.
5. Multithreaded Programming.
6. Implementation of basic CPU Scheduling algorithms like FCFS, SJF, Round Robin.
7. Implementation of Banker’s Algorithm.

Text Book:

Reference Books:

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.

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:

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

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

[B] IT Branch

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

*List of Program Electives is provided in Appendix-I

Semester-VI

[A] CSE Branch

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

*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

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>IT302</td>
<td>Software Project Management</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>IT362</td>
<td>Software Project Management Lab</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>IT304</td>
<td>Management Information System</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>-</td>
<td>Program Elective (PE3)*</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>-</td>
<td>Program Elective (PE4)*</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>-</td>
<td>Elective from other Branch of Engineering (EO1)*</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>-</td>
<td>Open Elective (OE1)§</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td>15</td>
<td>1</td>
<td>7</td>
<td>20</td>
</tr>
</tbody>
</table>

*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

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>L-T-P: C</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS301</td>
<td>Computer Networks</td>
<td>3-0-0: 3</td>
</tr>
</tbody>
</table>

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:
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.
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:

Reference Book:

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>L-T-P: C</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS361</td>
<td>Computer Networks Lab</td>
<td>0-0-2: 1</td>
</tr>
</tbody>
</table>

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 IIITV Network structure and basic networking hardware devices such as Hub, Switch, Router, Firewall, NIC, Modem, Gateway, etc. Write down and submit the difference between various networking devices.
2. Study various types of network cables and their usage. Construct the cross and straight cables using a crimping tool. Install and configure wired and wireless NIC.
3. Configure Host IP, Subnet Mask and Default Gateway in a System in LAN (TCP/IP Configuration). Transfer files between machines in LAN using FTP. Configuration and install a Print server in a LAN and share the printer.
4. Understand and implement basic networking commands such as Ping, Ipconfig, Traceroute, Arp,
Netstat, Whois, SSH, etc.
5. Implement standard error detection and correction methods.
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.

Reference: Laboratory instructions and manuals.

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>L-T-P: C</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS303</td>
<td>Software Engineering</td>
<td>3-0-0: 3</td>
</tr>
</tbody>
</table>

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
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:

Reference Books:

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 (NDFA), Equivalence between DFA and NDFA, 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 Bound 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.

Reference Books:

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.


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).


Text Books:

Reference Books:

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.
4. Cryptographic hash algorithm such as SHA-512.
5. Message authentication codes such as HMAC.
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.

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>L-T-P: C</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS391/ IT391</td>
<td>Design Project#</td>
<td>0-0-0: 3</td>
</tr>
</tbody>
</table>

*The Design Project is a Pass/Fail course. It is conducted during summer vacation between 4th and 5th semester. The evaluation of project work is carried out in 1st week of 5th semester and its grade will appear in the 5th semester grade sheet.*

**Scope of the Project:** To provide an opportunity to second year undergraduate students to apply individual efforts involving the knowledge of Information Technology, Computer Science, Embedded Systems or related areas in designing/ developing/ implementing complete systems.

Students are expected to carry out design project either individually or in a group of 2 – 4 students. The duration of the project work is between 6 – 8 weeks. Followings are the broad areas in which a student may work:
1. Software Development.
2. Software Maintenance.
5. Robotics.
Semester-VI: Courses and Contents

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>L-T-P: C</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS302</td>
<td>Artificial Intelligence</td>
<td>3-0-0: 3</td>
</tr>
</tbody>
</table>

**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:**

**References:**

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>L-T-P: C</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS362</td>
<td>Artificial Intelligence Lab</td>
<td>0-0-2: 1</td>
</tr>
</tbody>
</table>

**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.
5. Game Playing: Min-Max and Alpha-Beta Algorithm.
8. Neural Network.
9. Decision Tree.
10. Fuzzy Expert System

**Reference:** Laboratory instructions and handouts.

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>L-T-P: C</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS364</td>
<td>Introduction to Cryptography and Network Security</td>
<td>3-0-0: 3</td>
</tr>
</tbody>
</table>

**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:**

**Reference Books:**

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>L-T-P: C</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS364</td>
<td>Introduction to Cryptography and Network Security Lab</td>
<td>0-0-2: 1</td>
</tr>
</tbody>
</table>

**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.
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).
11. Implementation of elliptic curve arithmetic.
13. Secure socket layers (SSL)/Transport layer security (TLS).
15. Firewalls (e.g. iptables)

**Reference:** Laboratory instructions and handouts.

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>L-T-P: C</th>
</tr>
</thead>
<tbody>
<tr>
<td>IT302</td>
<td>Software Project Management</td>
<td>3-0-0: 3</td>
</tr>
</tbody>
</table>

**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:**


**Reference Book:**


<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>L-T-P: C</th>
</tr>
</thead>
<tbody>
<tr>
<td>IT362</td>
<td>Software Project Management Lab</td>
<td>0-0-3: 2</td>
</tr>
</tbody>
</table>

**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.
<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>L-T-P: C</th>
</tr>
</thead>
<tbody>
<tr>
<td>IT304</td>
<td>Management Information System</td>
<td>3-0-0: 3</td>
</tr>
</tbody>
</table>

**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:**

**Reference Book:**
**Course Structure: Fourth Year**

### Semester-VII

#### [A] CSE Branch

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS401</td>
<td>Introduction to Distributed and Parallel Computing</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>CS461</td>
<td>Introduction to Distributed and Parallel Computing Lab</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>-</td>
<td>Program Elective (PE5)*</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>-</td>
<td>Program Elective (PE6)*</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>-</td>
<td>Elective from other Branch of Engineering (EO2)*</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>-</td>
<td>Open Elective (OE2)§</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>CS491</td>
<td>Research/Industrial Internship</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td>15</td>
<td>0</td>
<td>4</td>
<td>20</td>
</tr>
</tbody>
</table>

*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

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>IT401</td>
<td>System Administration and Maintenance</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>IT461</td>
<td>System Administration and Maintenance Lab</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>-</td>
<td>Program Elective (PE5)*</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>-</td>
<td>Program Elective (PE6)*</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>-</td>
<td>Elective from other Branch of Engineering (EO2)*</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>-</td>
<td>Open Elective (OE2)§</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>IT491</td>
<td>Research/Industrial Internship</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td>14</td>
<td>0</td>
<td>5</td>
<td>20</td>
</tr>
</tbody>
</table>

*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

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS490/IT490</td>
<td>B. Tech. Project</td>
<td>0</td>
<td>0</td>
<td>36</td>
<td>18</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td>0</td>
<td>0</td>
<td>36</td>
<td>18</td>
</tr>
</tbody>
</table>

*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-VII: Courses and Contents

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>L-T-P: C</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS401</td>
<td>Introduction to Distributed and Parallel Computing</td>
<td>3-0-0: 3</td>
</tr>
</tbody>
</table>

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.


Naming: Naming Entities, Locating Mobile Entities, Removing Unreferenced Entities.

Synchronization: Clock Synchronization, Logical Clocks and Election Algorithms.


Text Books:

Objective: 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.
4. Java RMI.
5. Distributed Chat application.
6. Distributed Banking application.
7. Distributed File system application.
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.

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>L-T-P: C</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS461</td>
<td>Introduction to Distributed and Parallel Computing Lab</td>
<td>0-0-2: 1</td>
</tr>
</tbody>
</table>

Objective: 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.

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>L-T-P: C</th>
</tr>
</thead>
<tbody>
<tr>
<td>IT401</td>
<td>System Administration and Maintenance</td>
<td>2-0-0: 2</td>
</tr>
</tbody>
</table>

Objective: 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.


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


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:

Reference Books:

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>L-T-P: C</th>
</tr>
</thead>
<tbody>
<tr>
<td>IT461</td>
<td>System Administration and Maintenance Lab</td>
<td>0-0-3: 2</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>L-T-P: C</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS491/IT491</td>
<td>Research/Industrial Internship*</td>
<td>0-0-0: 3</td>
</tr>
</tbody>
</table>

*The Research/Industrial Internship is a Pass/Fail course. It is conducted during summer vacation between 6th and 7th semester. The evaluation of the internship is carried out in 1st week of 7th semester and its grade will appear in the 7th 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

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>L-T-P: C</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS490/IT490</td>
<td>B. Tech. Project</td>
<td>0-0-36: 18</td>
</tr>
</tbody>
</table>

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.
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.
3. Hardware Development/Implementation.

**Deliverables**: In general, the expected deliverables will include one or more of the followings
1. Product developed.
2. Code developed.
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 1st 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

<table>
<thead>
<tr>
<th>Program Verticals (PV)</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Artificial Intelligence and Data Analytics</td>
<td>PV1</td>
</tr>
<tr>
<td>Cyber Physical Systems</td>
<td>PV2</td>
</tr>
<tr>
<td>Security</td>
<td>PV3</td>
</tr>
<tr>
<td>Computational Science</td>
<td>PV4</td>
</tr>
</tbody>
</table>

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

Appendix-II

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

<table>
<thead>
<tr>
<th>Code</th>
<th>Course Name</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Digital Image Processing</td>
</tr>
<tr>
<td></td>
<td>Deep Learning</td>
</tr>
<tr>
<td></td>
<td>Information Theory Coding</td>
</tr>
<tr>
<td></td>
<td>Cognitive Science</td>
</tr>
<tr>
<td></td>
<td>Soft Computing</td>
</tr>
<tr>
<td></td>
<td>Advanced Image Processing</td>
</tr>
<tr>
<td></td>
<td>Graph Theory</td>
</tr>
<tr>
<td></td>
<td>Low Power Circuit Design</td>
</tr>
<tr>
<td></td>
<td>Real-time System</td>
</tr>
<tr>
<td></td>
<td>Nano Technology</td>
</tr>
<tr>
<td></td>
<td>VLSI Design</td>
</tr>
<tr>
<td></td>
<td>E-Commerce</td>
</tr>
</tbody>
</table>

Appendix-III

List of Open Electives (OE1 and OE2)

<table>
<thead>
<tr>
<th>Code</th>
<th>Course Name</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Operation Research</td>
</tr>
<tr>
<td></td>
<td>Network Flow Algorithms</td>
</tr>
<tr>
<td></td>
<td>Professional Ethics</td>
</tr>
<tr>
<td></td>
<td>Quantum Mechanics</td>
</tr>
<tr>
<td></td>
<td>Quantum Models in Science and Engineering</td>
</tr>
<tr>
<td></td>
<td>Introduction to Quantum Computation</td>
</tr>
<tr>
<td></td>
<td>Game Theory</td>
</tr>
<tr>
<td></td>
<td>Computational Physics</td>
</tr>
</tbody>
</table>