

### I YEAR

**I SEMESTER**

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Course Code</th>
<th>Course Title</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>BSC</td>
<td>Mathematics – I (Linear Algebra and Calculus)</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>ESC</td>
<td>Basic Electrical Engineering</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>BSC</td>
<td>Engineering Chemistry</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>HSMC</td>
<td>English</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>ESC</td>
<td>Basic Electrical Engineering Lab</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>2.5</td>
</tr>
<tr>
<td>6</td>
<td>BSC</td>
<td>Engineering Chemistry Lab</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1.5</td>
</tr>
<tr>
<td>7</td>
<td>HSMC</td>
<td>English Language and Communication Skills Lab</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>ESC</td>
<td>Engineering Workshop</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td><strong>Total Credits</strong></td>
<td></td>
<td>12</td>
<td>2</td>
<td>10</td>
<td>19</td>
</tr>
</tbody>
</table>

**II SEMESTER**

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Course Code</th>
<th>Course Title</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>BSC</td>
<td>Mathematics – II (Advanced Calculus)</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>BSC</td>
<td>Applied Physics</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>ESC</td>
<td>Programming for Problem Solving</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>ESC</td>
<td>Engineering Graphics</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>BSC</td>
<td>Applied Physics Lab</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1.5</td>
</tr>
<tr>
<td>6</td>
<td>ESC</td>
<td>Programming for Problem Solving Lab</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1.5</td>
</tr>
<tr>
<td></td>
<td><strong>Total Credits</strong></td>
<td></td>
<td>10</td>
<td>2</td>
<td>10</td>
<td>17</td>
</tr>
</tbody>
</table>
### JNTUH COLLEGE OF ENGINEERING HYDERABAD

(AUTONOMOUS)

B.TECH. FOUR YEAR DEGREE COURSE

(COMPUTER SCIENCE AND ENGINEERING)

COURSE STRUCTURE

### II YEAR

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Course Code</th>
<th>Course Title</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>CREDITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ESC</td>
<td>Analog &amp; Digital Electronics</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>PCC</td>
<td>Data Structures</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>PCC</td>
<td>Discrete Mathematics</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>PCC</td>
<td>Computer Organization &amp; Architecture</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>PCC</td>
<td>Object Oriented Programming</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>ESC</td>
<td>Analog &amp; Digital Electronics Lab</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1.5</td>
</tr>
<tr>
<td>7</td>
<td>PCC</td>
<td>Data Structures Lab</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1.5</td>
</tr>
<tr>
<td>8</td>
<td>PCC</td>
<td>Object Oriented Programming using C++ Lab</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1.5</td>
</tr>
<tr>
<td>9</td>
<td>PCC</td>
<td>IT Workshop Lab</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1.5</td>
</tr>
<tr>
<td></td>
<td><strong>TOTAL</strong></td>
<td></td>
<td>14</td>
<td>1</td>
<td>12</td>
<td>21</td>
</tr>
</tbody>
</table>

### I SEMESTER

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Course Code</th>
<th>Course Title</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>CREDITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>BSC</td>
<td>Computer Oriented Statistical Methods</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>HSMC</td>
<td>Business Economics &amp; Financial Analysis</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>PCC</td>
<td>Operating Systems</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>PCC</td>
<td>Database Management Systems</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>PCC</td>
<td>Design and Analysis of Algorithms</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>PCC</td>
<td>Operating Systems Lab</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1.5</td>
</tr>
<tr>
<td>7</td>
<td>PCC</td>
<td>Database Management Systems Lab</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1.5</td>
</tr>
<tr>
<td>8</td>
<td>PCC</td>
<td>Algorithms Lab using Java</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>MC</td>
<td>Environmental Science</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td><strong>TOTAL</strong></td>
<td></td>
<td>17</td>
<td>2</td>
<td>8</td>
<td>22</td>
</tr>
</tbody>
</table>

2
### III YEAR

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Course Code</th>
<th>Course Title</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>CREDITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PCC</td>
<td>Formal Languages &amp; Automata Theory</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>PCC</td>
<td>Software Engineering</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>PCC</td>
<td>Computer Networks</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>PCC</td>
<td>Artificial Intelligence</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>PEC</td>
<td>Professional Elective-I</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>PEC</td>
<td>Professional Elective -II</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>PCC</td>
<td>Software Engineering Lab</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1.5</td>
</tr>
<tr>
<td>8</td>
<td>PCC</td>
<td>Computer Networks Lab</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1.5</td>
</tr>
<tr>
<td>9</td>
<td>HSMC</td>
<td>Advanced English Communication Skills Lab</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>MC</td>
<td>Constitution of India</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td></td>
<td></td>
<td>20</td>
<td>0</td>
<td>8</td>
<td>22</td>
</tr>
</tbody>
</table>

### II SEMESTER

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Course Code</th>
<th>Course Title</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>CREDITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PCC</td>
<td>Machine Learning</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>PCC</td>
<td>Compiler Design</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>PCC</td>
<td>Web Technologies</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>PEC</td>
<td>Professional Elective - III</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>OEC</td>
<td>Open Elective-I</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>PCC</td>
<td>Machine Learning Lab using Python</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1.5</td>
</tr>
<tr>
<td>7</td>
<td>PCC</td>
<td>Compiler Design &amp; Web Technologies Lab</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>PCC</td>
<td>Professional Elective-III Lab</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1.5</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td></td>
<td></td>
<td>15</td>
<td>2</td>
<td>10</td>
<td>22</td>
</tr>
</tbody>
</table>
## IV YEAR

### I SEMESTER

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Course Code</th>
<th>Course Title</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>CREDITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PCC</td>
<td>Information Security</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>PCC</td>
<td>Big Data Analytics</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>PEC</td>
<td>Professional Elective -IV</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>PEC</td>
<td>Professional Elective -V</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>OEC</td>
<td>Open Elective-II</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>PCC</td>
<td>Big Data Analytics Lab</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>SI</td>
<td>Summer Project Internship*</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>PROJ</td>
<td>Project -I</td>
<td>0</td>
<td>0</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>9</td>
<td>PROJ</td>
<td>Seminar &amp; Technical Writing</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td></td>
<td></td>
<td>14</td>
<td>0</td>
<td>10</td>
<td>21</td>
</tr>
</tbody>
</table>

* To be carried out during the summer vacation between 6th and 7th semesters

## IV YEAR

### II SEMESTER

<table>
<thead>
<tr>
<th>S.No.</th>
<th>Course Code</th>
<th>Course Title</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>CREDITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>HSMC</td>
<td>Management Fundamentals for Engineers</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>PEC</td>
<td>Professional Elective -VI</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>OEC</td>
<td>Open Elective-III</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>PROJ</td>
<td>PROJECT -II</td>
<td>0</td>
<td>0</td>
<td>16</td>
<td>8</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td></td>
<td></td>
<td>9</td>
<td>0</td>
<td>16</td>
<td>17</td>
</tr>
</tbody>
</table>
Professional Elective Core (PEC) Courses for CSE

Professional Elective-I
1. Information Theory & Coding
2. Advanced Computer Architecture
3. Data Mining
4. Image Processing
5. Principles of Programming Languages

Professional Elective -II
1. Computer Graphics
2. Advanced Operating Systems
3. Informational Retrieval Systems
4. Advanced Databases
5. Natural Language Processing

Professional Elective -III
1. Concurrent Programming
2. Network Programming
3. Scripting Languages
4. Mobile Application Development
5. Software Testing Methodologies

Professional Elective -IV
1. Graph Theory
2. Embedded Systems
3. Semantic Web
4. Cloud Computing
5. Distributed Systems

Professional Elective -V
1. Advanced Algorithms
2. Mobile Computing
3. Soft Computing
4. Internet of Things
5. Software Process & Project Management

Professional Elective - VI
1. Computational Complexity
2. Neural Networks & Deep Learning
3. Cyber Forensics
4. Software Metrics & Measures
5. Ad hoc & Sensor Networks
MATHEMATICS-I
(LINEAR ALGEBRA AND CALCULUS)

I Year B.Tech. I-Semester

Pre-requisites: Mathematical Knowledge of 12th/Intermediate level

Objectives: To learn

1. Types of matrices and their properties.
2. Concept of a rank of the matrix and applying this concept to know the consistency and solving the system of linear equations.
3. Concept of Eigenvalues and Eigenvectors and to reduce the quadratic form to canonical form
4. Concept of Sequence.
5. Concept of nature of the series.
6. Geometrical approach to the mean value theorems and their application to the mathematical problems.
7. Evaluation of surface areas and volumes of revolutions of curves.
8. Evaluation of improper integrals using Beta and Gamma functions.
10. Finding maxima and minima of function of two and three variables.

Outcomes
After learning the contents of this paper the student must be able to

1. Write the matrix representation of a set of linear equations and to analyse the solution of the system of equations.
2. Find the Eigenvalues and Eigenvectors.
3. Reduce the quadratic form to canonical form using orthogonal transformations.
4. Analyse the nature of sequence and series.
5. Solve the applications on the mean value theorems.
6. Evaluate the improper integrals using Beta and Gamma functions.
7. Find the extreme values of functions of two variables with/without constraints.

UNIT-I: Matrices
Matrices: Types of Matrices, Symmetric; Hermitian; Skew-symmetric; Skew-Hermitian; orthogonal matrices; Unitary Matrices; Rrank of a matrix by Echelon form and Normal form, Inverse of Non-singular matrices by Gauss-Jordan method; System of linear equations; solving system of Homogeneous and Non-Homogeneous equations, Gauss elimination method; Gauss Seidel Iteration Method.

UNIT-II: Eigenvalues and Eigenvectors
Linear Transformation and Orthogonal Transformation: Eigenvalues and Eigenvectors and their properties: Diagonalization of a matrix; Cayley-Hamilton Theorem (without proof); finding inverse and power of a matrix by Cayley-Hamilton Theorem; Quadratic forms and Nature of the Quadratic Forms; Reduction of Quadratic form to canonical forms by Orthogonal Transformation.
UNIT-III: Sequences & Series
Sequence: Definition of a Sequence, limit; Convergent, Divergent and Oscillatory sequences.

Series: Convergent, Divergent and Oscillatory Series; Series of positive terms; Comparison test, p-test, D-Alembert’s ratio test; Raabe’s test; Cauchy’s Integral test; Cauchy’s root test; logarithmic test. Alternating series: Leibnitz test; Alternating Convergent series: Absolute and Conditionally Convergence.

UNIT-IV: Calculus
Mean value theorems: Rolle’s theorem, Lagrange’s Mean value theorem with their Geometrical Interpretation and applications, Cauchy’s Mean value Theorem. Taylor’s Series.

Applications of definite integrals to evaluate surface areas and volumes of revolutions of curves (Only in Cartesian coordinates), Definition of Improper Integral: Beta and Gamma functions and their applications.

UNIT-V: Multivariable calculus (Partial Differentiation and applications)
Definitions of Limit and continuity.
Partial Differentiation; Euler’s Theorem; Total derivative; Jacobian; Functional dependence & independence, Maxima and Minima of functions of two variables and three variables using method of Lagrange multipliers.

Textbooks:

References:
BASIC ELECTRICAL ENGINEERING

I Year B.Tech. I-Semester

Pre-requisites: --

Objectives

1. To introduce the concepts of electrical circuits and its components
2. To understand magnetic circuits, DC circuits and AC single phase & three phase circuits
3. To study and understand the different types of DC/AC machines and Transformers.
4. To import the knowledge of various electrical installations.
5. To introduce the concept of power, power factor and its improvement.

Outcomes

1. To analyze and solve electrical circuits using network laws and theorems.
2. To understand and analyze basic Electric and Magnetic circuits
3. To study the working principles of Electrical Machines
4. To introduce components of Low Voltage Electrical Installations

UNIT - I
D.C. CIRCUITS

UNIT - II
A.C. 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 in series R-L-C circuit. Three-phase balanced circuits, voltage and current relations in star and delta connections.

UNIT - III
TRANSFORMERS
Ideal and practical transformer, equivalent circuit, losses in transformers, regulation and efficiency. Auto-transformer and three-phase transformer connections.

UNIT - IV
ELECTRICAL MACHINES

UNIT - V
ELECTRICAL INSTALLATIONS
Components of LT Switchgear: Switch Fuse Unit (SFU), MCB, ELCB, MCCB, Types of Wires and Cables, Earthing. Types of Batteries, Important Characteristics for Batteries. Elementary calculations for energy consumption, power factor improvement and battery backup.
Textbooks:


References:

ENGINEERING CHEMISTRY

I Year B.Tech. I-Semester

Objectives
1. To bring adaptability to the concepts of chemistry and to acquire the required skills to become a perfect engineer.
2. To impart the basic knowledge of atomic, molecular and electronic modifications which makes the student to understand the technology based on them.
3. To acquire the knowledge of electrochemistry, corrosion and water treatment which are essential for the Engineers and in industry.
4. To acquire the skills pertaining to spectroscopy and to apply them for medical field etc.
5. To impart then knowledge of stereochemistry and synthetic aspects useful for understanding reaction pathways

Outcomes
1. The knowledge of atomic, molecular and electronic changes, band theory related to conductivity.
2. The required principles and concepts of electrochemistry, corrosion and in understanding the problem of water and its treatments.
3. The required skills to get clear concepts on basic spectroscopy and application to medical field etc.
4. The knowledge and configurational and conformational analysis of molecules and reaction mechanisms.

UNIT - I
Molecular structure and Theories of Bonding:
Atomic and Molecular orbitals. Linear Combination of Atomic Orbitals (LCAO), molecular orbitals of diatomic molecules, molecular orbital energy level diagrams of N₂, O₂ and NO molecules. Bond order.

UNIT - II
Water and its treatment:
UNIT – III
Electrochemistry and corrosion:
Electrochemical cells – electrode potential, standard electrode potential, types of electrodes – Calomel, Quinhydrone and glass electrode. Nernst equation Determination of pH of a solution by using quinhydrone and glass electrode. Electrochemical series and its applications. Numerical problems. Potentiometric titrations. Batteries – Primary (Lithium cell) and secondary batteries (Lead – acid storage battery and Lithium ion battery).


UNIT - IV
Stereochemistry, Reaction Mechanism and synthesis of drug molecules:
Representation of 3-dimensional structures, Isomers-Structural and stereoisomers, Enantiomers, diastereomers, symmetry and chirality. optical activity Absolute configuration. Conformational alanalysis of n- butane.

Reduction reactions: Reduction of carbonyl compounds using LiAlH_4 & NaBH_4. Structure, synthesis and pharmaceutical applications of Paracetamol and Aspirin.

UNIT - V
Spectroscopic techniques and applications:

Textbooks:

References:
1. Physical Chemistry, by P.W. Atkins
2. Engineering Chemistry (NPTEL Web-book), by B.L. Tembe, Kamaluddin and M.S. Krishnan
4. Fundamentals of Molecular Spectroscopy, by C.N. Banwell
INTRODUCTION
In view of the growing importance of English as a tool for global communication and the consequent emphasis on training students to acquire language skills, the syllabus of English has been designed to develop linguistic, communicative and critical thinking competencies of Engineering students.

In English classes, the focus should be on the skills development in the areas of vocabulary, grammar, reading and writing. For this, the teachers should use the prescribed text for detailed study. The students should be encouraged to read the texts leading to reading comprehension and different passages may be given for practice in the class. The time should be utilized for working out the exercises given after each excerpt, and for supplementing the exercises with authentic materials of a similar kind, for example, newspaper articles, advertisements, promotional material etc. *The focus in this syllabus is on skill development in the areas of Vocabulary, Grammar, Reading and Writing Skills, fostering ideas and practice of language skills in various contexts.*

Learning Objectives
The course will help students to
1. Improve the language proficiency of students in English with an emphasis on Vocabulary, Grammar, Reading and Writing skills.
2. Equip students to study academic subjects more effectively and critically using the theoretical and practical components of English syllabus.
3. Develop study skills and communication skills in formal and informal situations.

Outcomes
Students should be able to
1. Use English Language effectively in spoken and written forms.
2. Comprehend the given texts and respond appropriately.
3. Communicate confidently in various contexts and different cultures.
4. The student will acquire basic proficiency in English including reading and listening comprehension, writing, and speaking skills.

SYLLABUS
(Note: As the syllabus of English given in AICTE Model Curriculum-2018 for B.Tech First Year is Open-ended, it is required to prepare teaching/learning materials by the teachers collectively in the form of handouts based on the needs of the students in their respective colleges for effective teaching/learning and timesaving in the class.)

Unit –I
**Vocabulary Building:** The Concept of Word Formation --The Use of Prefixes and Suffixes.
**Grammar:** Identifying Common Errors in Writing with Reference to Articles and Prepositions.
**Reading:** Reading and Its Importance- Techniques for Effective Reading.
**Basic Writing Skills:** Sentence Structures -Use of Phrases and Clauses in Sentences- Importance of Proper Punctuation- Techniques for Writing Precisely – **Paragraph writing** – Types, Structures and Features of a Paragraph - Creating Coherence.

**Unit –II**

**Vocabulary:** Synonyms and Antonyms.

**Grammar:** Identifying Common Errors in Writing with Reference to Noun-pronoun Agreement and Subject-verb Agreement.

**Reading:** Improving Comprehension Skills – Techniques for Good Comprehension.


**Unit –III**

**Vocabulary:** Acquaintance with Prefixes and Suffixes from Foreign Languages in English to form Derivatives-Words from Foreign Languages and their Use in English.

**Grammar:** Identifying Common Errors in Writing with Reference to Misplaced Modifiers and Tenses.

**Reading:** Sub-skills of Reading- Skimming and Scanning

**Writing:** Writing Introduction and Conclusion - Essay Writing.

**Unit –IV**

**Vocabulary:** Standard Abbreviations in English

**Grammar:** Redundancies and Clichés in Oral and Written Communication.

**Reading:** Comprehension- Intensive Reading and Extensive Reading.

**Writing:** Writing Practices—Précis Writing.

**Unit –V**

**Vocabulary:** Technical Vocabulary and their usage

**Grammar:** Common Errors in English

**Reading:** Reading Comprehension-Exercises for Practice

**Writing:** Technical Reports- Introduction – Characteristics of a Report – Categories of Reports Formats- Structure of Reports (Manuscript Format) -Types of Reports - Writing a Report.

**Note:** Listening and Speaking skills which are given under Unit-6 are covered in the syllabus of ELCS Lab Course.

**References:**

BASIC ELECTRICAL ENGINEERING LAB

I Year B.Tech. I-Semester

Pre-requisites: Basic Electrical Engineering

Objectives

1. To analyze a given network by applying various electrical laws and network theorem
2. To know the response of electrical circuits for different excitations
3. To calculate, measure and know the relation between basic electrical parameters.
4. To analyze the performance characteristics of DC and AC electrical machines

Outcomes

1. Get an exposure to basic electrical laws.
2. Understand the response of different types of electrical circuits to different excitations.
3. Understand the measurement, calculation and relation between the basic electrical parameters
4. Understand the basic characteristics of transformers and electrical machines.

List of experiments/demonstrations:

1. Verification of Ohms Law
2. Verification of KVL and KCL
3. Transient Response of Series RL and RC circuits for DC excitation
4. Transient Response of RLC Series circuit for DC excitation
5. Resonance in series RLC circuit
6. Calculations and Verification of Impedance and Current of RL, RC and RLC series circuits
8. Load Test on Single Phase Transformer (Calculate Efficiency and Regulation)
9. Three Phase Transformer: Verification of Relationship between Voltages and Currents (Star-Delta, Delta-Delta, Delta-star, Star-Star)
10. Measurement of Active and Reactive Power in a balanced Three-phase circuit
11. Performance Characteristics of a Separately/Self Excited DC Shunt/Compound Motor
12. Torque-Speed Characteristics of a Separately/Self Excited DC Shunt/Compound Motor
13. Performance Characteristics of a Three-phase Induction Motor
14. Torque-Speed Characteristics of a Three-phase Induction Motor
15. No-Load Characteristics of a Three-phase Alternator

Textbooks:


References:

ENGINEERING CHEMISTRY LAB

I Year B.Tech. I-Semester

Objectives

The chemistry laboratory course consists of experiments related to the principles of chemistry required to the engineering student. The course will make the student to learn:

1. Estimation of hardness and chloride content in water to check its suitability for drinking purpose.
2. To determine the rate constant of reactions from concentrations as an function of time.
3. The measurement of physical properties like adsorption and viscosity.
4. To synthesize the drug molecules and check the purity of organic molecules by thin layer chromatographic (TLC) technique.

Outcomes

1. Determination of parameters like hardness and chloride content in water.
2. Estimation of rate constant of a reaction from concentration – time relationships.
3. Determination of physical properties like adsorption and viscosity.
4. Calculation of $R_f$ values of some organic molecules by TLC technique.

List of Experiments:

1. Determination of total hardness of water by complexometric method using EDTA
2. Estimation of Fe$^{3+}$ by Dichrometry.
3. Estimation of an HCl by Conductometric titrations
4. Estimation of Acetic acid by Conductometric titrations
5. Estimation of HCl by Potentiometric titrations
6. Estimation of Fe$^{2+}$ by Potentiometry using KMnO$_4$
7. Determination of rate constant of acid catalysed hydrolysis of methyl acetate
8. Synthesis of Aspirin and Paracetamol
9. Thin layer chromatography calculation of $R_f$ values. eg ortho and para nitro phenols
10. Determination of acid value of coconut oil
11. Verification of freundlich adsorption isotherm-adsorption of acetic acid on charcoal
12. Determination of viscosity of Coconut oil and ground nut oil by using Ostwald’s viscometer.
14. Determination of partition coefficient of acetic acid between n-butanol and water.

References:

1. Senior practical physical chemistry, B.D. Khosla, A. Gulati and V. Garg (R. Chand & Co., Delhi)
2. An introduction to practical chemistry, K.K. Sharma and D. S. Sharma (Vikas publishing, N. Delhi)
ENGLISH LANGUAGE AND COMMUNICATION SKILLS LAB

I Year B.Tech. I-Semester

<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

The Language Lab focuses on the production and practice of sounds of language and familiarizes the students with the use of English in everyday situations both in formal and informal contexts.

Objectives
1. To facilitate computer-assisted multi-media instruction enabling individualized and independent language learning
2. To sensitize students to the nuances of English speech sounds, word accent, intonation and rhythm
3. To bring about a consistent accent and intelligibility in students’ pronunciation of English by providing an opportunity for practice in speaking
4. To improve the fluency of students in spoken English and neutralize their mother tongue influence
5. To train students to use language appropriately for public speaking and interviews

Learning Outcomes
Students will be able to attain
1. Better understanding of nuances of English language through audio-visual experience and group activities
2. Neutralization of accent for intelligibility
3. Speaking skills with clarity and confidence which in turn enhances their employability skills

Syllabus
English Language and Communication Skills Lab (ELCS) shall have two parts:
   a. Computer Assisted Language Learning (CALL) Lab
   b. Interactive Communication Skills (ICS) Lab

Listening Skills
Objectives
1. To enable students, develop their listening skills so that they may appreciate its role in the LSRW skills approach to language and improve their pronunciation
2. To equip students with necessary training in listening so that they can comprehend the speech of people of different backgrounds and regions

Students should be given practice in listening to the sounds of the language, to be able to recognize them and find the distinction between different sounds, to be able to mark stress and recognize and use the right intonation in sentences.
   • Listening for general content
   • Listening to fill up information
   • Intensive listening
   • Listening for specific information
Speaking Skills
Objectives
1. To involve students in speaking activities in various contexts
2. To enable students express themselves fluently and appropriately in social and professional contexts
   • Oral practice: Just A Minute (JAM) Sessions
   • Describing objects/situations/people
   • Role play – Individual/Group activities

➢ The following course content is prescribed for the English Language and Communication Skills Lab based on Unit-6 of AICTE Model Curriculum 2018 for B.Tech First English. As the syllabus is very limited, it is required to prepare teaching/learning materials by the teachers collectively in the form of handouts based on the needs of the students in their respective colleges for effective teaching/learning and timesaving in the Lab)

Exercise – I
CALL Lab:
Understand: Listening Skill- Its importance – Purpose- Process- Types- Barriers of Listening.
ICS Lab:
Understand: Communication at Work Place- Spoken vs. Written language.

Exercise – II
CALL Lab:
Practice: Basic Rules of Word Accent - Stress Shift - Weak Forms and Strong Forms in Context.
ICS Lab:

Exercise - III
CALL Lab:
Understand: Intonation-Errors in Pronunciation-the Influence of Mother Tongue (MTI).
Practice: Common Indian Variants in Pronunciation – Differences in British and American Pronunciation.
ICS Lab:
Understand: How to make Formal Presentations.
Practice: Formal Presentations.

Exercise – IV
CALL Lab:
Understand: Listening for General Details.
Practice: Listening Comprehension Tests.
ICS Lab:
Understand: Public Speaking – Exposure to Structured Talks.
Practice: Making a Short Speech – Extempore.
Exercise – V

CALL Lab:
*Understand:* Listening for Specific Details.
*Practice:* Listening Comprehension Tests.

ICS Lab:
1. Introduction to Interview Skills.
2. Common errors in speaking.

Minimum Requirement of infrastructural facilities for ELCS Lab:

1. **Computer Assisted Language Learning (CALL) Lab:**
   - The Computer Assisted Language Learning Lab has to accommodate 40 students with 40 systems, with one Master Console, LAN facility and English language learning software for self-study by students.

System Requirement (Hardware component):
- Computer network with LAN facility (minimum 40 systems with multimedia) with the following specifications:
  i) Computers with Suitable Configuration
  ii) High Fidelity Headphones

2. **Interactive Communication Skills (ICS) Lab:**
   - The Interactive Communication Skills Lab: A Spacious room with movable chairs and audio-visual aids with a Public-Address System, a LCD and a projector etc.

*************
ENGINEERING WORKSHOP

I Year B.Tech. I-Semester

Pre-requisites: Practical skill

Objectives

1. To Study of different hand operated power tools, uses and their demonstration.
2. To gain a good basic working knowledge required for the production of various engineering products.
3. To provide hands on experience about use of different engineering materials, tools, equipment and processes those are common in the engineering field.
4. To develop a right attitude, team working, precision and safety at work place.
5. It explains the construction, function, use and application of different working tools, equipment and machines.
6. To study commonly used carpentry joints.
7. To have practical exposure to various welding and joining processes.
8. Identify and use marking out tools, hand tools, measuring equipment and to work to prescribed tolerances.
9. To understand the computer hardware and practice the Assembly of computer parts.
10. To practice the process of Installation of operating system windows.

Outcomes

1. Practice on manufacturing of components using workshop trades including pluming, fitting, carpentry, foundry, house wiring and welding.
2. Identify and apply suitable tools for different trades of Engineering processes including drilling, material removing, measuring, chiseling.
3. Apply basic electrical engineering knowledge for house wiring practice.

I. TRADES FOR EXERCISES:
   (Any six trades from the following with minimum of two exercises in each trade)
   1. Carpentry – 2 Lectures
   2. Fitting- 1Lecture
   3. Tin-Smithy- 1Lecture
   4. Black Smithy-1Lecture
   5. House-wiring-1Lecture
   6. Foundry- 2 Lectures
   7. Plumbing-1Lecture

II. Trades for Demonstration & Exposure
   1. Demonstration of power tools -1 Lecture
   2. Welding – 2 Lecture
   3. Machine Shop -2 Lectures

III. IT Workshop I: Computer hardware, identification of parts, Disassembly, Assembly of computer to working condition, simple diagnostic exercises.
**IT Workshop II:** Installation of operating system windows and linux simple diagnostic exercises.

**Textbooks:**

1. Workshop Practice by B.L.Juneja Cengage Learning
MATHEMATICS-II  
(ADVANCED CALCULUS)  

I Year B.Tech. II-Semester

L  T  P  C
3  1  0  4

Pre-requisites: Mathematical Knowledge of 12\textsuperscript{th} / Intermediate level

Objectives: To learn
1. Methods of solving the differential equations of first and higher order. 
2. Evaluation of multiple integrals and their applications 
3. The physical quantities involved in engineering field related to vector valued functions 
4. The basic properties of vector valued functions and their applications to line, surface and volume integrals

Outcomes: 
After learning the contents of this paper the student must be able to
1. Identify whether the given differential equation of first order is exact or not
2. Solve higher differential equation and apply the concept of differential equation to real world problems
3. Evaluate the multiple integrals and apply the concept to find areas, volumes, centre of mass and gravity for cubes, sphere and rectangular parallel piped
4. Evaluate the line, surface and volume integrals and converting them from one to another

UNIT-I: First Order ODE
Exact, linear and Bernoulli’s equations; Applications: Newton’s law of cooling, Law of natural growth and decay; Equations not of first degree: equations solvable for \( p \), equations solvable for \( y \), equations solvable for \( x \) and Clairaut’s type.

UNIT-II: Ordinary Differential Equations of Higher Order
Second order linear differential equations with constant coefficients: Non-Homogeneous terms of the type \( e^{ax}, \sin ax, \cos ax \), polynomials in \( x \), \( e^{ax}\), \( x \), \( x^2 \); method of variation of parameters; Equations reducible to linear ODE with constant coefficients: Legendre’s equation, Cauchy-Euler equation.

UNIT-III: Multivariable Calculus (Integration)
Evaluation of Double Integrals (Cartesian and polar coordinates); change of order of integration (only Cartesian form); Evaluation of Triple Integrals: Change of variables (Cartesian to polar) for double and (Cartesian to Spherical and Cylindrical polar coordinates) for triple integrals. Applications: Areas (by double integrals) and volumes (by double integrals and triple integrals), Centre of mass and Gravity (constant and variable densities) by double and triple integrals (applications involving cubes, sphere and rectangular parallel piped).

UNIT-IV: Vector Differentiation
UNIT-V: Vector Integration
Line, Surface and Volume Integrals. Theorems of Green, Gauss and Stokes (without proofs) and their applications.

Textbooks:

References:
APPLIED PHYSICS

I Year B.Tech. II-Semester

Objectives
1. Understand basic principle of quantum mechanics
2. Gain the knowledge of carrier concentration and recombination process of semiconductor materials.
3. Learn about various types of optoelectronic devices
4. Know about Various types of lasers and significance of optical fibers in communication system
5. Learn about material properties like dielectrics and magnetic materials.

Outcomes
1. Analyze the wave particle duality and about energy levels and uncertainty principle
2. Evaluate the mobility of charge carrier concentration of a given semiconductor material.
3. Justify how the graded index optical fiber is more efficient than step index optical fiber in fiber optic communication system.
4. To learn about working of LED, solar cell and photo detector
5. Gain the knowledge and applications of dielectric and magnetic materials

UNIT-I: Quantum Mechanics
Introduction to quantum mechanics, Black body radiation, Planck’s law, Photoelectric effect, Compton effect, Wave-particle duality, de-Broglie hypothesis, Davisson and Germer experiment, Heisenberg’s uncertainty principle, Born’s interpretation of the wave function, Schrodinger’s time independent wave equation, Particle in one dimensional box.

UNIT-II: Semiconductor Physics

UNIT-III: Optoelectronic Devices

UNIT- IV: Lasers and Fibre Optics

UNIT-V: Dielectric and Magnetic Properties of Materials
Dielectrics Properties: Introduction, Types of polarizations, Electronic, Ionic (quantitative) and Oriental Polarizations and calculation of their polarizabilities, Internal fields in a solid- Clausius-Mossotti relation, Applications of dielectric materials.
**Magnetic Properties:** Introduction, Bohr magneton, classification of dia, para and ferro magnetic materials on the basis of magnetic moment, Explanation of Hysteresis curve based on domain theory of ferromagnetism, Soft and hard magnetic materials, Properties of anti-ferro and ferri magnetic materials.

**Text Books:**

**References:**
1. Richard Robinett, Quantum Mechanics.
3. Online Course: “Optoelectronic Materials and Devices” by Monica Katiyar and Deepak Gupta on NPTEL.
PROGRAMMING FOR PROBLEM SOLVING

I Year B.Tech. II-Semester

Objectives

1. To learn the fundamentals of computers.
2. To understand the various steps in Program development.
3. To learn the syntax and semantics of C Programming Language.
4. To learn the usage of structured programming approach in solving problems.

Outcomes

1. Write algorithms and to draw flowcharts for solving problems.
2. Translate the algorithms/flowcharts to programs (in C language).
3. Code and test a given logic in C programming language.
4. Formulate simple algorithms for arithmetic and logical problems.
5. Decompose a problem into functions and to develop modular reusable code.
6. Use arrays, pointers, strings and structures to formulate algorithms and programs.
7. Searching and sorting problems.

UNIT-I:
INTRODUCTION TO C LANGUAGE – Background, Simple C programs, Identifiers, Basic data types, Variables, Constants, Input / Output, Operators. Expressions, Precedence and Associatively, Expression Evaluation, Type conversions, Bit wise operators, Statements, Simple C Programming examples.

UNIT-II:
STATEMENTS – if and switch statements, Repetition statements – while, for, do-while statements, Loop examples, other statements related to looping – break, continue, go to, Simple C Programming examples.
DESIGNING STRUCTURED PROGRAMS- Functions, basics, user defined functions, inter function communication, Scope, Storage classes-auto, register, static, extern, scope rules, type qualifiers, recursion-recurive functions, Preprocessor commands, example C programs

UNIT-III:
ARRAYS AND STRINGS – Concepts, using arrays in C, inter function communication, array applications, two – dimensional arrays, multidimensional arrays, C program examples. Concepts, C Strings, String Input / Output functions, arrays of strings, string manipulation functions, string / data conversion, C program examples.

UNIT-IV:
POINTERs – Introduction (Basic Concepts), Pointers for inter function communication, pointers to pointers, compatibility, memory allocation functions, array of pointers, programming applications, pointers to void, pointers to functions, command –line arguments.
INPUT AND OUTPUT – Concept of a file, streams, standard input / output functions, formatted input / output functions, text files and binary files, file input / output operations, file status functions (error handling), C program examples.

UNIT-V:
DERIVED TYPES – Structures – Declaration, definition and initialization of structures, accessing structures, nested structures, arrays of structures, structures and functions, pointers to structures, self referential structures, unions, typedef, bit fields, enumerated types, C programming examples.
SORTING AND SEARCHING – Selection sort, Bubble sort, Insertion sort, Linear search and Binary search methods.

Textbooks:

3. The C Programming Language, B.W. Kernighan and Dennis M. Ritchie, PHI/Pearson Education

References:

3. C Programming & Data Structures, P. Dey, M Ghosh R Thereja, Oxford University Press
ENGINEERING GRAPHICS

I Year B.Tech. II-Semester

Pre-requisites: Nil

Objectives
1. To provide basic concepts in engineering drawing.
2. To impart knowledge about standard principles of orthographic projection of objects.
3. To draw sectional views and pictorial views of solids.

Outcomes
1. Preparing working drawings to communicate the ideas and information.
2. Read, understand and interpret engineering drawings.

UNIT - I
INTRODUCTION TO ENGINEERING DRAWING
Principles of Engineering Graphics and their Significance, Conic Sections including the Rectangular
Hyperbola – General method only. Cycloid, Epicycloid and Hypocycloid, Involute. Scales – Plain,
Diagonal and Vernier Scales.

UNIT - II
ORTHOGRAPHIC PROJECTIONS:
Principles of Orthographic Projections – Conventions – Projections of Points and Lines, Projections of
Plane regular geometric figures.—Auxiliary Planes.

UNIT - III
Projections of Regular Solids – Auxiliary Views.

UNIT - IV
Sections or Sectional views of Right Regular Solids – Prism, Cylinder, Pyramid, Cone – Auxiliary views –
Sections of Sphere. Development of Surfaces of Right Regular Solids – Prism, Cylinder, Pyramid and Cone

UNIT - V
ISOMETRIC PROJECTIONS :
Principles of Isometric Projection – Isometric Scale – Isometric Views – Conventions – Isometric Views of
Lines, Plane Figures, Simple and Compound Solids – Isometric Projection of objects having non- isometric
lines. Isometric Projection of Spherical Parts. Conversion of Isometric Views to Orthographic Views and
Vice-versa – Conventions

Auto CAD: Basic principles only

Textbooks:
1. Engineering Drawing, N.D. Bhatt, Charotar
2. Engineering Drawing and Graphics, Rane and Shah, Pearson Education.

References:
1. A Text Book of Engineering Drawing, Dhawan R K, S. Chand
APPLIED PHYSICS LAB

I Year B.Tech. II-Semester

<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1.5</td>
</tr>
</tbody>
</table>

Objectives
1. To provide an experimental foundation for the theoretical concepts introduced in the lectures.
2. To teach how to make careful experimental observations and how to think about and draw conclusions from such data.
3. To help students understand the role of direct observation in physics and to distinguish between inferences based on theory and the outcomes of experiments.

Outcomes
1. Make careful experimental observations and draw conclusions from such data.
2. Distinguish between inferences based on theory and the outcomes of experiments.
3. Write a technical report which communicates scientific information in a clear and concise manner.

List of experiments:

1. Energy gap of P-N junction diode: To determine the energy gap of a semiconductor diode.
2. Solar Cell: To study the V-I Characteristics of solar cell.
3. Light emitting diode: Plot V-I and P-I characteristics of light emitting diode.
4. Stewart – Gee’s experiment: Determination of magnetic field along the axis of a current carrying coil.
5. Hall Effect: To determine Hall co-efficient of a given semiconductor.
6. Photoelectric effect: To determine work function of a given material.
7. LASER: To study the characteristics of LASER sources.
8. Losses in Optical fibre: To determine the bending losses of Optical fibres.
9. LCR Circuit: To determine the resonant frequency and quality factor of LCR Circuit
10. R-C Circuit: To determine the time constant of R-C circuit.
11. PIN photo diode: To study the V-I & P-I characteristics of PIN photo diode detector.
12. Zener diode: To study the V-I Characteristics of Zener diode.

Note: Any 8 experiments are to be performed by each student
PROGRAMMING FOR PROBLEM SOLVING LAB

I Year B.Tech. II-Semester

Objectives
1. To learn the fundamentals of computers.
2. To understand the various steps in Program development.
3. To learn the syntax and semantics of C Programming Language.
4. To learn the usage of structured programming approach in solving problems.

Outcomes
1. Write algorithms and to draw flowcharts for solving problems.
2. Translate the algorithms/flowcharts to programs (in C language).
3. Code and test a given logic in C programming language.
4. Formulate simple algorithms for arithmetic and logical problems.
5. Decompose a problem into functions and to develop modular reusable code.
6. Use arrays, pointers, strings and structures to formulate algorithms and programs.
7. Searching and sorting problems.

Week 1:
1. Write a C program to find the sum of individual digits of a positive integer.
2. Fibonacci sequence is defined as follows: the first and second terms in the sequence are 0 and 1.
   Subsequent terms are found by adding the preceding two terms in the sequence.
3. Write a C program to generate the first n terms of the sequence.
4. Write a C program to generate all the prime numbers between 1 and n, where n is a value supplied by
   the user.
5. Write a C program to find the roots of a quadratic equation.

Week 2:
6. Write a C program to find the factorial of a given integer.
7. Write a C program to find the GCD (greatest common divisor) of two given integers.
8. Write a C program to solve Towers of Hanoi problem.
9. Write a C program, which takes two integer operands and one operator from the user, performs the
   operation and then prints the result. (Consider the operators +, -, *, /, % and use Switch Statement)

Week 3:
10. Write a C program to find both the largest and smallest number in a list of integers.
11. Write a C program that uses functions to perform the following:
    i) Addition of Two Matrices    ii) Multiplication of Two Matrices

Week 4:
12. Write a C program that uses functions to perform the following operations:
    i) To insert a sub-string in to a given main string from a given position.
    ii) To delete n Characters from a given position in a given string.
13. Write a C program to determine if the given string is a palindrome or not
14. Write a C program that displays the position or index in the string S where the string T begins, or – 1 if
    S doesn’t contain T.
15. Write a C program to count the lines, words and characters in a given text.

**Week 5:**
16. Write a C program to generate Pascal’s triangle.
17. Write a C program to construct a pyramid of numbers.
18. Write a C program to read in two numbers, x and n, and then compute the sum of this geometric progression:
   \[ 1 + x + x^2 + x^3 + \ldots + x^n \]
   For example: if n is 3 and x is 5, then the program computes 1+5+25+125.
   Print x, n, the sum
   Perform error checking.
   For example, the formula does not make sense for negative exponents – if n is less than 0.
   Have your program print an error message if n<0, then go back and read in the next pair of numbers of without computing the sum. Are any values of x also illegal? If so, test for them too.

**Week 6:**
19. 2’s complement of a number is obtained by scanning it from right to left and complementing all the bits after the first appearance of a 1. Thus 2’s complement of 11100 is 00100. Write a C program to find the 2’s complement of a binary number.
20. Write a C program to convert a Roman numeral to its decimal equivalent.

**Week 7:**
21. Write a C program that uses functions to perform the following operations:
   i) Reading a complex number
   ii) Writing a complex number
   iii) Addition of two complex numbers
   iv) Multiplication of two complex numbers
   (Note: represent complex number using a structure.)

**Week 8:**
22. . i) Write a C program which copies one file to another.
      ii) Write a C program to reverse the first n characters in a file.
      (Note: The file name and n are specified on the command line.)
23. . i) Write a C program to display the contents of a file.
      ii) Write a C program to merge two files into a third file (i.e., the contents of the first file followed by those of the second are put in the third file)

**Week 9:**
24. Write a C program that implements the following sorting methods to sort a given list of integers in ascending order
   i) Bubble sort     ii) Selection sort    iii) Insertion sort
Week 10:
25. Write C programs that use both recursive and non recursive functions to perform the following searching
   Operations for a Key value in a given list of integers:
   i) Linear search        ii) Binary search

Textbooks:
3. The C Programming Language, B.W. Kernighan and Dennis M.Ritchie, PHI/Pearson Education

References:
3. C Programming & Data Structures, P. Dey, M Ghosh R Thereja, Oxford University Press
ANALOG & DIGITAL ELECTRONICS

II Year B.Tech. I-Semester

<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>

Objectives
1. To introduce components such as diodes, BJTs and FETs.
2. To know the applications of components.
3. To give understanding of various types of amplifier circuits
4. To learn basic techniques for the design of digital circuits and fundamental concepts used in the
design of digital systems.
5. To understand the concepts of combinational logic circuits and sequential circuits.

Outcomes
1. Know the characteristics of various components.
2. Understand the utilization of components.
3. Design and analyze small signal amplifier circuits.
4. Postulates of Boolean algebra and to minimize combinational functions
5. Design and analyze combinational and sequential circuits
6. Known about the logic families and realization of logic gates.

UNIT - I: Diodes and applications
Junction diode characteristics: Open circuited p-n junction, p-n junction as a rectifier, V-I characteristics,
effect of temperature, diode resistance, diffusion capacitance, diode switching times, breakdown diodes,
Tunnel diodes, photo diode, LED.

Diode Applications - clipping circuits, comparators, Half wave rectifier, Full wave rectifier, rectifier with
capacitor filter.

UNIT - II: BJTs
Transistor characteristics: The junction transistor, transistor as an amplifier, CB, CE, CC configurations,
comparison of transistor configurations, the operating point, self-bias or Emitter bias, bias compensation,
thermal runaway and stability, transistor at low frequencies, CE amplifier response, gain bandwidth
product, Emitter follower, RC coupled amplifier, two cascaded CE and multi stage CE amplifiers.

UNIT - III: FETs and Digital Circuits
FETs: JFET, V-I characteristics, MOSFET, low frequency CS and CD amplifiers, CS and CD amplifiers.

Digital Circuits: Digital (binary) operations of a system, OR gate, AND gate, NOT, EXCLUSIVE OR gate,
De Morgan Laws, NAND and NOR DTL gates, modified DTL gates, HTL and TTL gates, output stages,
RTL and DCTL, CMOS, Comparison of logic families.

UNIT - IV: Combinational logic circuits
Basic Theorems and Properties of Boolean Algebra, Canonical and Standard Forms, Digital Logic Gates,
The Map Method, Product-of-Sums Simplification, Don’t-Care Conditions, NAND and NOR Implementation, Exclusive-OR Function, Binary Adder-Subtractor, Decimal Adder, Binary Multiplier,
Magnitude Comparator, Decoders, Encoders, Multiplexers.
UNIT - V: Sequential logic circuits
Sequential Circuits, Storage Elements: Latches and flip flops, Analysis of Clocked Sequential Circuits, State Reduction and Assignment, Shift Registers, Ripple Counters, Synchronous Counters, Random-Access Memory, Read-Only Memory.

Textbooks:

References:
DATA STRUCTURES

II Year B.Tech. I-Semester

Prerequisites
1. A course on “Programming for Problem Solving “

Objectives
1. Exploring basic data structures such as stacks and queues.
2. Introduces a variety of data structures such as hash tables, search trees, tries, heaps, graphs.
3. Introduces sorting and pattern matching algorithms

Outcomes
1. Ability to select the data structures that efficiently model the information in a problem.
2. Ability to assess efficiency trade-offs among different data structure implementations or combinations.
3. Implement and know the application of algorithms for sorting and pattern matching.
4. Design programs using a variety of data structures, including hash tables, binary and general tree structures, search trees, tries, heaps, graphs, and AVL-trees.

UNIT - I
Introduction to Data Structures, abstract data types, Linear list – singly linked list implementation, insertion, deletion and searching operations on linear list, Stacks-Operations, array and linked representations of stacks, stack applications, Queues-operations, array and linked representations.

UNIT - II
Dictionaries: linear list representation, skip list representation, operations - insertion, deletion and searching.

Hash table representation: hash functions, collision resolution-separate chaining, open addressing-linear probing, quadratic probing, double hashing, rehashing, extendible hashing.

UNIT - III

UNIT - IV
Graphs: Graph Implementation Methods. Graph Traversal Methods.
Sortings: Heap Sort, External Sorting- Model for external sorting, Merge Sort.

UNIT - V
Pattern matching and Tries: Pattern matching algorithms-Brute force, the Boyer –Moore algorithm, the Knuth-Morris-Pratt algorithm, Standard Tries, Compressed Tries, Suffix tries.
Textbooks:

References:
2. Introduction to data structures in C, Ashok Kamthane, 1st Edition, PEARSON.
DISCRETE MATHEMATICS

II Year B.Tech. I-Semester

<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td>4</td>
</tr>
</tbody>
</table>

Prerequisites
1. An understanding of Mathematics in general is sufficient.

Objectives
1. Introduces the elementary discrete mathematics for computer science and engineering.
2. Topics include formal logic notation, methods of proof, induction, sets, relations, graph theory, permutations and combinations, counting principles; recurrence relations and generating functions.

Outcomes
1. Ability to understand and construct precise mathematical proofs
2. Ability to use logic and set theory to formulate precise statements
3. Ability to analyze and solve counting problems on finite and discrete structures
4. Ability to describe and manipulate sequences
5. Ability to apply graph theory in solving computing problems

UNIT - I
The Foundations: Logic and Proofs
Propositional Logic, Applications of Propositional Logic, Propositional Equivalence, Predicates and Quantifiers, Nested Quantifiers, Rules of Inference, Introduction to Proofs, Proof Methods and Strategy.

UNIT - II
Basic Structures, Sets, Functions, Sequences, Sums, Matrices and Relations
Sets, Functions, Sequences & Summations, Cardinality of Sets and Matrices
Relations, Relations and Their Properties, n-ary Relations and Their Applications, Representing Relations, Closures of Relations, Equivalence Relations, Partial Orderings.

UNIT - III
Algorithms, Induction and Recursion

Induction and Recursion
Mathematical Induction, Strong Induction and Well-Ordering, Recursive Definitions and Structural Induction, Recursive Algorithms, Program Correctness.

UNIT - IV
Discrete Probability and Advanced Counting Techniques
An Introduction to Discrete Probability. Probability Theory, Bayes’ Theorem, Expected Value and Variance.

Advanced Counting Techniques
UNIT - V

Graphs
Graphs and Graph Models, Graph Terminology and Special Types of Graphs, Representing Graphs and Graph Isomorphism, Connectivity, Euler and Hamilton Paths, Shortest-Path Problems, Planar Graphs, Graph Coloring.

Trees
Introduction to Trees, Applications of Trees, Tree Traversal, Spanning Trees, Minimum Spanning Trees.

Textbook:


References:

1. Discrete Mathematical Structures with Applications to Computer Science-J.P. Tremblay and R. Manohar, TMH,
COMPUTER ORGANIZATION & ARCHITECTURE

II Year B.Tech. I-Semester

<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>

Prerequisites: No prerequisites

Co-requisite: A Course on “Digital Logic Design and Microprocessors”

Objectives

1. The purpose of the course is to introduce principles of computer organization and the basic architectural concepts.
2. It begins with basic organization, design, and programming of a simple digital computer and introduces simple register transfer language to specify various computer operations.
3. Topics include computer arithmetic, instruction set design, microprogrammed control unit, pipelining and vector processing, memory organization and I/O systems, and multiprocessors

Outcomes

1. Understand the basics of instructions sets and their impact on processor design.
2. Demonstrate an understanding of the design of the functional units of a digital computer system.
3. Evaluate cost performance and design trade-offs in designing and constructing a computer processor including memory.
4. Design a pipeline for consistent execution of instructions with minimum hazards.
5. Recognize and manipulate representations of numbers stored in digital computers

UNIT - I


Register Transfer Language and Micro operations: Register Transfer language, Register Transfer, Bus and memory transfers, Arithmetic Micro operations, logic micro operations, shift micro operations, Arithmetic logic shift unit.

Basic Computer Organization and Design: Instruction codes, Computer Registers Computer instructions, Timing and Control, Instruction cycle, Memory Reference Instructions, Input – Output and Interrupt.

UNIT - II

Micro Programmed Control: Control memory, Address sequencing, micro program example, design of control unit.

Central Processing Unit: General Register Organization, Instruction Formats, Addressing modes, Data Transfer and Manipulation, Program Control.

UNIT - III

Data Representation: Data types, Complements, Fixed Point Representation, Floating Point Representation.

UNIT - IV

Memory Organization: Memory Hierarchy, Main Memory, Auxiliary memory, Associate Memory, Cache Memory.

UNIT - V
Reduced Instruction Set Computer: CISC Characteristics, RISC Characteristics.

Pipeline and Vector Processing: Parallel Processing, Pipelining, Arithmetic Pipeline, Instruction Pipeline, RISC Pipeline, Vector Processing, Arrey Processor.

Multi Processors: Characteristics of Multiprocessors, Interconnection Structures, Interprocessor arbitration, Interprocessor communication and synchronization, Cache Coherence.

Textbook:

References:
OBJECT ORIENTED PROGRAMMING

II Year B.Tech. I-Semester

Prerequisites
1. A course on “Programming for Problem Solving ”

Objectives
1. Introduces object oriented programming concepts using the C++ language.
2. Introduces the principles of data abstraction, inheritance and polymorphism;
3. Introduces the principles of virtual functions and polymorphism
4. Introduces handling formatted I/O and unformatted I/O
5. Introduces exception handling

Outcomes
1. Able to develop programs with reusability
2. Develop programs for file handling
3. Handle exceptions in programming
4. Develop applications for a range of problems using object-oriented programming techniques

UNIT - I
Object Oriented thinking - Different paradigms for problem solving, need for OOP paradigm, differences between OOP and Procedure oriented programming, Overview of OOP concepts- Abstraction, Encapsulation, Inheritance and Polymorphism.

C++ Basics: Structure of a C++ program, Data types, Declaration of variables, Expressions, Operators, Operator Precedence, Evaluation of expressions, Type conversions, Pointers, Arrays, Pointers and Arrays, Strings, Structures, References.

Flow control statement- if, switch, while, for, do, break, continue, goto statements.

Functions - Scope of variables, Parameter passing, Default arguments, inline functions, Recursive functions, Pointers to functions.

Dynamic memory allocation and deallocation operators-new and delete, Preprocessor directives.

UNIT - II
C++ Classes And Data Abstraction: Class definition, Class structure, Class objects, Class scope, this pointer, Friends to a class, Static class members, Constant member functions, Constructors and Destructors, Dynamic creation and destruction of objects, Data abstraction, ADT and information hiding.

UNIT - III
Inheritance: Defining a class hierarchy, Different forms of inheritance, Defining the Base and Derived classes, Access to the base class members, Base and Derived class construction, Destructors, Virtual base class.

Virtual Functions And Polymorphism: Static and Dynamic binding, virtual functions, Dynamic binding through virtual functions, Virtual function call mechanism, Pure virtual functions, Abstract classes, Implications of polymorphic use of classes, Virtual destructors.
UNIT - IV
C++ I/O: I/O using C functions, Stream classes hierarchy, Stream I/O, File streams and String streams, Overloading operators, Error handling during file operations, Formatted I/O.

UNIT-V
Exception Handling: Benefits of exception handling, Throwing an exception, The try block, Catching an exception, Exception objects, Exception specifications, Stack unwinding, Rethrowing an exception, Catching all exceptions.

Textbooks:

References:
ANALOG & DIGITAL ELECTRONICS LAB

II Year B.Tech. I-Semester

L T P C
0 0 3 1.5

Objectives
1. To introduce components such as diodes, BJTs and FETs.
2. To know the applications of components.
3. To give understanding of various types of amplifier circuits
4. To learn basic techniques for the design of digital circuits and fundamental concepts used in the
   design of digital systems.
5. To understand the concepts of combinational logic circuits and sequential circuits.

Outcomes
1. Know the characteristics of various components.
2. Understand the utilization of components.
3. Design and analyze small signal amplifier circuits.
4. Postulates of Boolean algebra and to minimize combinational functions
5. Design and analyze combinational and sequential circuits
6. Known about the logic families and realization of logic gates.

List of Experiments
1. Full Wave Rectifier with & without filters
2. Common Emitter Amplifier Characteristics
3. Common Base Amplifier Characteristics
4. Common Source amplifier Characteristics
5. Measurement of h-parameters of transistor in CB, CE, CC configurations
6. Input and Output characteristics of FET in CS configuration
7. Realization of Boolean Expressions using Gates
8. Design and realization logic gates using universal gates
9. generation of clock using NAND / NOR gates
10. Design a 4 – bit Adder / Subtractor
11. Design and realization a Synchronous and Asynchronous counters using flip-flops
12. Realization of logic gates using DTL, TTL, ECL, etc.,

Textbooks:
1. Integrated Electronics: Analog and Digital Circuits and Systems, Jacob Millman, Christos Halkias

References:
DATA STRUCTURES LAB

II Year B.Tech. I-Semester

L T P C
0 0 3 1.5

Prerequisites:
1. A Course on “Programming for problem solving”

Objectives
1. It covers various concepts of C programming language
2. It introduces searching and sorting algorithms
3. It provides an understanding of data structures such as stacks and queues.

Outcomes
1. Ability to develop C programs for computing and real life applications using basic elements like control statements, arrays, functions, pointers and strings, and data structures like stacks, queues and linked lists.
2. Ability to Implement searching and sorting algorithms

List of Experiments

1. Write a program that uses functions to perform the following operations on singly linked list.: i) Creation ii) Insertion iii) Deletion iv) Traversal
2. Write a program that uses functions to perform the following operations on doubly linked list.: i) Creation ii) Insertion iii) Deletion iv) Traversal
3. Write a program that uses functions to perform the following operations on circular linked list.: i) Creation ii) Insertion iii) Deletion iv) Traversal
4. Write a program that implement stack (its operations) using i) Arrays ii) Pointers
5. Write a program that implement Queue (its operations) using i) Arrays ii) Pointers
6. Write a program that implements the following sorting methods to sort a given list of integers in ascending order i) Bubble sort ii) Selection sort iii) Insertion sort
7. Write a program that use both recursive and non recursive functions to perform the following searching operations for a Key value in a given list of integers: i) Linear search ii) Binary search
8. Write a program to implement the tree traversal methods
9. Write a program to implement the graph traversal methods

Textbooks:

References:
2. Introduction to data structures in C, Ashok Kamthane, 1st Edition, PEARSON.
OBJECT ORIENTED PROGRAMMING USING C++ LAB

II Year B.Tech. I-Semester                                                                                                           L   T   P   C
                                                                                                                                    0  0  3  1.5

Prerequisites: A course on “Programming for Problem Solving ”

Objectives
1. Introduces object oriented programming concepts using the C++ language.
2. Introduces the principles of data abstraction, inheritance and polymorphism;
3. Introduces the principles of virtual functions and polymorphism
4. Introduces handling formatted I/O and unformatted I/O
5. Introduces exception handling

Outcomes
1. Ability to develop applications for a range of problems using object-oriented programming techniques

List of Experiments

1. Program: Write a C++ Program to display names, roll no’s, and grades of 3 students who have appeared in the examination. Declare the class of name, roll no’s and grade. Create an array of class objects. Read and display the contents of the array.
2. Program: Write a C++ program to declare struct. Initialize and display contents of member variables.
3. Program: Write a C++ program to declare a class. Declare pointer to class. Initialize and display the contents of the class member.
4. Program: Given that an EMPLOYEE class contains following members: data members: Employee number, Employee name, Basic, DA, IT, Net Salary and print data members. Write a C++ program to read the data of N employee and compute Net salary of each employee (DA=52% of Basic and Income Tax (IT) =30% of the gross salary).
5. Program: Write a C++ to illustrate the concepts of console I/O operations.
6. Write a C++ program to use scope resolution operator. Display the various values of the same variables declared at different scope levels.
7. Program: Write a C++ program to allocate memory using new operator.
8. Write a C++ program to create multilevel inheritance. (Hint:classes A1,A2, A3)
9. Write a C++ program to create an array of pointers. Invoke functions using array objects.
10. Write a C++ program to use pointer for both base and derived classes and call the member function. Use Virtual keyword.

Textbooks:

References:
IT WORKSHOP LAB

II Year B.Tech. I-Semester

<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1.5</td>
</tr>
</tbody>
</table>

Objectives:

The IT Workshop for engineers is a training lab course spread over 60 hours. The modules include training on PC Hardware, Internet & World Wide Web and Productivity tools including Word, Excel, Power Point and Publisher.

PC Hardware introduces the students to a personal computer and its basic peripherals, the process of assembling a personal computer, installation of system software like MS Windows, Linux and the required device drivers. In addition hardware and software level troubleshooting process, tips and tricks would be covered. The students should work on working PC to disassemble and assemble to working condition and install Windows and Linux on the same PC. Students are suggested to work similar tasks in the Laptop scenario wherever possible. Internet & World Wide Web module introduces the different ways of hooking the PC on to the internet from home and workplace and effectively usage of the internet. Usage of web browsers, email, newsgroups and discussion forums would be covered. In addition, awareness of cyber hygiene, i.e., protecting the personal computer from getting infected with the viruses, worms and other cyber attacks would be introduced. Productivity tools module would enable the students in crafting professional word documents, excel spread sheets, power point presentations and personal web sites using the Microsoft suite of office tools and LaTeX. (Recommended to use Microsoft office 2007 in place of MS Office 2003)

PC Hardware

Task 1: Identify the peripherals of a computer, components in a CPU and its functions. Draw the block diagram of the CPU along with the configuration of each peripheral and submit to your instructor.

Task 2: Every student should disassemble and assemble the PC back to working condition. Lab instructors should verify the work and follow it up with a Viva. Also students need to go through the video which shows the process of assembling a PC. A video would be given as part of the course content.

Task 3: Every student should individually install MS windows on the personal computer. Lab instructor should verify the installation and follow it up with a Viva.

Task 4: Every student should install Linux on the computer. This computer should have windows installed. The system should be configured as dual boot with both windows and Linux. Lab instructors should verify the installation and follow it up with a Viva.

Task 5: Hardware Troubleshooting: Students have to be given a PC which does not boot due to improper assembly or defective peripherals. They should identify the problem and fix it to get the computer back to working condition. The work done should be verified by the instructor and followed up with a Viva.

Task 6: Software Troubleshooting: Students have to be given a malfunctioning CPU due to system software problems. They should identify the problem and fix it to get the computer back to working condition. The work done should be verified by the instructor and followed up with a Viva.

Internet & World Wide Web

Task1: Orientation & Connectivity Boot Camp: Students should get connected to their Local Area Network and access the Internet. In the process they configure the TCP/IP setting. Finally students should
demonstrate, to the instructor, how to access the websites and email. If there is no internet connectivity preparations need to be made by the instructors to simulate the WWW on the LAN.

**Task 2: Web Browsers, Surfing the Web:** Students customize their web browsers with the LAN proxy settings, bookmarks, search toolbars and pop up blockers. Also, plug-ins like Macromedia Flash and JRE for applets should be configured.

**Task 3: Search Engines & Netiquette:** Students should know what search engines are and how to use the search engines. A few topics would be given to the students for which they need to search on Google. This should be demonstrated to the instructors by the student.

**Task 4: Cyber Hygiene:** Students would be exposed to the various threats on the internet and would be asked to configure their computer to be safe on the internet. They need to first install an antivirus software, configure their personal firewall and windows update on their computer. Then they need to customize their browsers to block pop ups, block active x downloads to avoid viruses and/or worms.

**LaTeX and Word**

**Task 1 – Word Orientation:** The mentor needs to give an overview of LaTeX and Microsoft (MS) office 2007/ equivalent (FOSS) tool word: Importance of LaTeX and MS office 2007/ equivalent (FOSS) tool Word as word Processors, Details of the four tasks and features that would be covered in each, Using LaTeX and word – Accessing, overview of toolbars, saving files, Using help and resources, rulers, format painter in word.

**Task 2: Using LaTeX and Word** to create project certificate. Features to be covered:– Formatting Fonts in word, Drop Cap in word, Applying Text effects, Using Character Spacing, Borders and Colors, Inserting Header and Footer, Using Date and Time option in both LaTeX and Word.

**Task 3: Creating project abstract** Features to be covered:- Formatting Styles, Inserting table, Bullets and Numbering, Changing Text Direction, Cell alignment, Footnote, Hyperlink, Symbols, Spell Check, Track Changes.

**Task 4 : Creating a Newsletter :** Features to be covered:- Table of Content, Newspaper columns, Images from files and clipart, Drawing toolbar and Word Art, Formatting Images, Textboxes, Paragraphs and Mail Merge in word.

**Excel**

**Excel Orientation:** The mentor needs to tell the importance of MS office 2007/ equivalent (FOSS) tool Excel as a Spreadsheet tool, give the details of the four tasks and features that would be covered in each. Using Excel – Accessing, overview of toolbars, saving excel files, Using help and resources.

**Task 1: Creating a Scheduler -** Features to be covered: Gridlines, Format Cells, Summation, auto fill, Formatting Text

**Task 2 : Calculating GPA -** .Features to be covered:- Cell Referencing, Formulae in excel – average, std.deviation, Charts, Renaming and Inserting worksheets, Hyper linking, Count function, LOOKUP/VLOOKUP

**Task 3: Performance Analysis -** Features to be covered:- Split cells, freeze panes, group and outline, Sorting, Boolean and logical operators, Conditional formatting
LaTeX and MS/equivalent (FOSS) tool Power Point

Task1: Students will be working on basic power point utilities and tools which help them create basic power point presentation. Topic covered during this week includes :- PPT Orientation, Slide Layouts, Inserting Text, Word Art, Formatting Text, Bullets and Numbering, Auto Shapes, Lines and Arrows in both LaTeX and Powerpoint. Students will be given model power point presentation which needs to be replicated (exactly how it’s asked).

Task 2: Second week helps students in making their presentations interactive. Topic covered during this week includes: Hyperlinks, Inserting –Images, Clip Art, Audio, Video, Objects, Tables and Charts.

Task 3: Concentrating on the in and out of Microsoft power point and presentations in LaTeX. Helps them learn best practices in designing and preparing power point presentation. Topic covered during this week includes: - Master Layouts (slide, template, and notes), Types of views (basic, presentation, slide slotter, notes etc), and Inserting – Background, textures, Design Templates, Hidden slides.

References:

1. Comdex Information Technology course tool kit, Vikas Gupta, WILEY Dreamtech
3. Introduction to Information Technology, ITL Education Solutions limited, Pearson Education.
4. PC Hardware and A+Handbook, Kate J. Chase, PHI (Microsoft)
5. LaTeX Companion, Leslie Lamport, PHI/Pearson.
COMPUTER ORIENTED STATISTICAL METHODS

II Year B.Tech. II-Semester

Pre-requisites: Mathematics courses of first year of study.

Objectives

To learn

1. The theory of Probability, and probability distributions of single and multiple random variables
2. The sampling theory and testing of hypothesis and making inferences

Outcomes

After learning the contents of this paper the student must be able to

1. Apply the concepts of probability and distributions to some case studies
2. Correlate the material of one unit to the material in other units
3. Resolve the potential misconceptions and hazards in each topic of study.

UNIT - I: Probability

Sample Space, Events, Counting Sample Points, Probability of an Event, Additive Rules, Conditional Probability, Independence, and the Product Rule, Bayes’ Rule,


UNIT - II: Mathematical Expectation

Mean of a Random Variable, Variance and Covariance of Random Variables, Means and Variances of Linear Combinations of Random Variables, Chebyshev’s Theorem.


UNIT - III: Continuous Probability Distributions

Continuous Uniform Distribution, Normal Distribution, Areas under the Normal Curve, Applications of the Normal Distribution, Normal Approximation to the Binomial, Gamma and Exponential Distributions.

Fundamental Sampling Distributions: Random Sampling, Some Important Statistics, Sampling Distributions, Sampling Distribution of Means and the Central Limit Theorem, Sampling Distribution of \( S^2 \), \( t \)-Distribution, F-Distribution.

UNIT - IV: Estimation & Tests of Hypotheses


UNIT - V: Stochastic Processes And Markov Chains

Textbooks:

References:
2. Sheldon M Ross, Probability and statistics for Engineers and scientists, academic press.
BUSINESS ECONOMICS & FINANCIAL ANALYSIS

II Year B.Tech. II-Semester

<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>

Objective

1. To prepare engineering students to analyze cost/revenue/financial data and to make economic and financial analysis in decision making process and to examine the performance of companies engaged in engineering.

Outcome

1. To perform and evaluate present and future worth of the alternate projects and to appraise projects by using traditional and DCF Methods. To carry out cost benefit analysis of projects and to calculate BEP of different alternative projects.

UNIT - I

UNIT - II

UNIT - III
Production, Cost, Market Structures & Pricing:

UNIT - IV

UNIT - V
Textbooks:
3. Accounting, Jain and Narang, Kalyani Publishers.
OPERATING SYSTEMS

II Year B.Tech. II-Semester

Prerequisites
1. A course on “Computer Programming and Data Structures”
2. A course on “Computer Organization and Architecture”

Objectives
1. Provide an introduction to operating system concepts (i.e., processes, threads, scheduling, synchronization, deadlocks, memory management, file and I/O subsystems and protection)
2. Introduce the issues to be considered in the design and development of operating system
3. Introduce basic Unix commands, system call interface for process management, interprocess communication and I/O in Unix

Outcomes
1. Will be able to control access to a computer and the files that may be shared
2. Demonstrate the knowledge of the components of computer and their respective roles in computing.
3. Ability to recognize and resolve user problems with standard operating environments.
4. Gain practical knowledge of how programming languages, operating systems, and architectures interact and how to use each effectively.

UNIT - I

UNIT - II
System call interface for process management-fork, exit, wait, waitpid, exec

UNIT - III
Deadlocks - System Model, Deadlocks Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, and Recovery from Deadlock.
Interprocess Communication Mechanisms: IPC between processes on a single computer system, IPC between processes on different systems, using pipes, FIFOs, message queues, shared memory.

UNIT - IV

UNIT - V
Textbooks:


References:

2. Operating System A Design Approach-Crowley,TMH.
4. Unix programming environment, Kernighan and Pike, PHI / Pearson Education
DATABASE MANAGEMENT SYSTEMS

II Year B.Tech. II-Semester

L T P C
3 0 0 3

Prerequisites
1. A course on “Data Structures”

Objectives
1. To understand the basic concepts and the applications of database systems.
2. To master the basics of SQL and construct queries using SQL.
3. Topics include data models, database design, relational model, relational algebra, transaction control, concurrency control, storage structures and access techniques.

Outcomes
1. Gain knowledge of fundamentals of DBMS, database design and normal forms
2. Master the basics of SQL for retrieval and management of data.
3. Be acquainted with the basics of transaction processing and concurrency control.
4. Familiarity with database storage structures and access techniques

UNIT - I
Database System Applications: A Historical Perspective, File Systems versus a DBMS, the Data Model, Levels of Abstraction in a DBMS, Data Independence, Structure of a DBMS

Introduction to Database Design: Database Design and ER Diagrams, Entities, Attributes, and Entity Sets, Relationships and Relationship Sets, Additional Features of the ER Model, Conceptual Design With the ER Model

UNIT - II
Introduction to the Relational Model: Integrity constraint over relations, enforcing integrity constraints, querying relational data, logical data base design, introduction to views, destroying/altering tables and views.

Relational Algebra, Tuple relational Calculus, Domain relational calculus.

UNIT - III
SQL: Queries, Constraints, Triggers: form of basic SQL query, UNION, INTERSECT, and EXCEPT, Nested Queries, aggregation operators, NULL values, complex integrity constraints in SQL, triggers and active data bases.

Schema refinement: Problems caused by redundancy, decompositions, problems related to decomposition, reasoning about functional dependencies, FIRST, SECOND, THIRD normal forms, BCNF, lossless join decomposition, multi-valued dependencies, FOURTH normal form, FIFTH normal form.

UNIT - IV
UNIT - V

Textbooks:

References:
2. SQL The Complete Reference, James R. Groff, Paul N. Weinberg, 3rd Edition,
3. Oracle for Professionals, The X Team, S.Shah and V. Shah, SPD.
DESIGN AND ANALYSIS OF ALGORITHMS

II Year B.Tech. II-Semester

Prerequisites
1. A course on “Computer Programming and Data Structures”
2. A course on “Advanced Data Structures”

Objectives
1. Introduces the notations for analysis of the performance of algorithms.
2. Introduces the data structure disjoint sets.
3. Describes major algorithmic techniques (divide-and-conquer, backtracking, dynamic programming, greedy, branch and bound methods) and mention problems for which each technique is appropriate;
4. Describes how to evaluate and compare different algorithms using worst-, average-, and best-case analysis.
5. Explains the difference between tractable and intractable problems, and introduces the problems that are P, NP and NP complete.

Outcomes
1. Ability to analyze the performance of algorithms
2. Ability to choose appropriate data structures and algorithm design methods for a specified application
3. Ability to understand how the choice of data structures and the algorithm design methods impact the performance of programs

UNIT - I
Introduction: Algorithm, Performance Analysis-Space complexity, Time complexity, Asymptotic Notations- Big oh notation, Omega notation, Theta notation and Little oh notation.
Divide and conquer: General method, applications-Binary search, Quick sort, Merge sort, Strassen’s matrix multiplication.

UNIT - II
Disjoint Sets: Disjoint set operations, union and find algorithms
Backtracking: General method, applications, n-queen’s problem, sum of subsets problem, graph coloring

UNIT - III
Dynamic Programming: General method, applications- Optimal binary search trees, 0/1 knapsack problem, All pairs shortest path problem, Traveling sales person problem, Reliability design.

UNIT - IV

UNIT - V
Branch and Bound: General method, applications - Travelling sales person problem, 0/1 knapsack problem - LC Branch and Bound solution, FIFO Branch and Bound solution.
NP-Hard and NP-Complete problems: Basic concepts, non deterministic algorithms, NP - Hard and NP-Complete classes, Cook’s theorem.
Textbook:

References:
OPERATING SYSTEMS LAB
(Using UNIX/LINUX)

II Year B.Tech. II-Semester

<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1.5</td>
</tr>
</tbody>
</table>

Prerequisites
1. A course on “Programming for Problem Solving”
2. A course on “Computer Organization and Architecture”

Co-requisite
1. A course on “Operating Systems”

Objectives
1. To provide an understanding of the design aspects of operating system concepts through simulation
2. Introduce basic Unix commands, system call interface for process management, interprocess communication and I/O in Unix

Outcomes
1. Simulate and implement operating system concepts such as scheduling, deadlock management, file management and memory management.
2. Able to implement C programs using Unix system calls

List of Experiments
1. Write C programs to simulate the following CPU Scheduling algorithms
   a) FCFS  
   b) SJF  
   c) Round Robin  
   d) priority
2. Write programs using the I/O system calls of UNIX/LINUX operating system (open, read, write, close, fcntl, seek, stat, opendir, readdir)
3. Write a C program to simulate Bankers Algorithm for Deadlock Avoidance and Prevention.
4. Write a C program to implement the Producer – Consumer problem using semaphores using UNIX/LINUX system calls.
5. Write C programs to illustrate the following IPC mechanisms
   a) Pipes  
   b) FIFOs  
   c) Message Queues  
   d) Shared Memory
6. Write C programs to simulate the following memory management techniques
   a) Paging  
   b) Segmentation

Textbooks:

References:
2. Operating System A Design Approach-Crowley, TMH.
4. Unix programming environment, Kernighan and Pike, PHI. / Pearson Education
5. Unix Internals The New Frontiers, U.Vahalia, Pearson Education
DATABASE MANAGEMENT SYSTEMS LAB

II Year B.Tech. II-Semester

<table>
<thead>
<tr>
<th></th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1.5</td>
</tr>
</tbody>
</table>

Co-requisites
1. Co-requisite of course “Database Management Systems”

Objectives
1. Introduce ER data model, database design and normalization
2. Learn SQL basics for data definition and data manipulation

Outcomes
1. Design database schema for a given application and apply normalization
2. Acquire skills in using SQL commands for data definition and data manipulation.
3. Develop solutions for database applications using procedures, cursors and triggers

List of Experiments
1) Concept design with E-R Model
2) Relational Model
3) Normalization
4) Practicing DDL commands
5) Practicing DML commands
6) Querying (using ANY, ALL, IN, Exists, NOT EXISTS, UNION, INTERSECT, Constraints etc.)
7) Queries using Aggregate functions, GROUP BY, HAVING and Creation and dropping of Views.
8) Triggers (Creation of insert trigger, delete trigger, update trigger)
9) Procedures
10) Usage of Cursors

Textbooks:

References:
2. SQL The Complete Reference, James R. Groff, Paul N. Weinberg, 3rd Edition,
3. Oracle for Professionals, The X Team, S.Shah and V. Shah, SPD.
ALGORITHMS LAB USING JAVA

II Year B.Tech. II-Semester

L  T  P  C
0  1  2  2

Prerequisites

1. A Course on “Data Structures”
2. A Course on “Objected Oriented Programming through Java”

Objectives

1. It covers various concepts of java programming language
2. It introduces searching and sorting algorithms
3. It introduces the feasible and optimal solutions by using the different design methods

Outcomes

1. Develop the feasible and optimal solutions by using Greedy and dynamic programming.
2. Able to design the searching algorithms

List Of Programs:

1. Write a program to implement n-Queen’s problem
2. Write a program to implement Optimal Binary Search Tree
3. Write a program to implement 0/1 Knapsack problem by using Dynamic Programming
4. Write a program to implement Greedy Knapsack problem
5. Write a program to implement Prim’s minimum cost spanning tree by using Greedy Method
6. Write a program to implement Kruskal’s minimum cost spanning tree by using Greedy Method
7. Write a program to implement Job sequencing with deadlines by using Greedy Method
8. Write a program to implement Single source shortest path problem by using Greedy Method

Textbooks:


References:

ENVIRONMENTAL SCIENCE

II Year B.Tech. II-Semester

Pre-Requisites: NIL

Objectives
1. Creating the awareness about environmental problems among students.
2. Imparting basic knowledge about the environment and its allied problems.
3. Developing an attitude of concern for the environment.
4. Motivating students to participate in environment protection and environment improvement.

Outcomes
At the end of the course, it is expected that students will be able to:
1. Identify and analyze environmental problems as well as the risks associated with these problems
2. Understand what it is to be a steward in the environment
3. Studying how to live their lives in a more sustainable manner

UNIT - I
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 – Forest resources – Use and over – exploitation, deforestation, case studies – Timber extraction – Mining, dams and other effects on forest and tribal people – Water resources – Use and over utilization of surface and ground water – Floods, drought, conflicts over water, dams – benefits and problems - Mineral resources: Use and exploitation, environmental effects of extracting and using mineral resources, case studies.

UNIT – II
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 ecosystem:
a. Forest ecosystem
b. Grassland ecosystem
c. Desert ecosystem
d. Aquatic ecosystems (ponds, streams, lakes, rivers, oceans, estuaries)

UNIT – III
UNIT – IV
Environmental Pollution and control:
Environmental Pollution: Classification of pollution, Air Pollution: Primary and secondary pollutants, Automobile and Industrial pollution, Ambient air quality standards.
Water pollution: Sources and types of pollution, drinking water quality standards. Soil Pollution: Sources and types, Impacts of modern agriculture, degradation of soil.

UNIT – V
SOCIAL ISSUES AND THE ENVIRONMENT: From Unsustainable to Sustainable development - Urban problems related to energy - Water conservation, rain water harvesting, watershed management - Resettlement and rehabilitation of people; its problems and concerns. Case Studies - Environmental ethics:


Textbooks:
1. Textbook of Environmental Studies for Undergraduate Courses by Erach Bharucha for University Grants Commission., Universities Press
2. Environmental Studies by R. Rajagopalan, Oxford University Press.

References:
1. Textbook of Environmental Sciences and Technology by M. Anji Reddy, BS Publication.
FORMAL LANGUAGES AND AUTOMATA THEORY

III Year B.Tech. I-Semester

Objectives

1. To provide introduction to some of the central ideas of theoretical computer science from the perspective of formal languages.
2. To introduce the fundamental concepts of formal languages, grammars and automata theory.
3. Classify machines by their power to recognize languages.
4. Employ finite state machines to solve problems in computing.
5. To understand deterministic and non-deterministic machines.
6. To understand the differences between decidability and undecidability.

Outcomes

1. Able to understand the concept of abstract machines and their power to recognize the languages.
2. Able to employ finite state machines for modeling and solving computing problems.
3. Able to design context free grammars for formal languages.
4. Able to distinguish between decidability and undecidability.
5. Able to gain proficiency with mathematical tools and formal methods.

UNIT - I

Introduction to Finite Automata: Structural Representations, Automata and Complexity, the Central Concepts of Automata Theory – Alphabets, Strings, Languages, Problems.


Deterministic Finite Automata: Definition of DFA, How A DFA Process Strings, The language of DFA, Conversion of NFA with €-transitions to NFA without €-transitions. Conversion of NFA to DFA, Moore and Melay machines

UNIT - II


Pumping Lemma for Regular Languages: Statement of the pumping lemma, Applications of the Pumping Lemma.

Closure Properties of Regular Languages: Closure properties of Regular languages, Decision Properties of Regular Languages, Equivalence and Minimization of Automata.

UNIT - III


Push Down Automata: Definition of the Pushdown Automaton, the Languages of a PDA, Equivalence of PDA's and CFG's, Acceptance by final state, Acceptance by empty stack, Deterministic Pushdown Automata. From CFG to PDA, From PDA to CFG.
UNIT - IV
Normal Forms for Context-Free Grammars: Eliminating useless symbols, Eliminating $\epsilon$-Productions.
Chomsky Normal form, Greibech Normal form.

Pumping Lemma for Context-Free Languages: Statement of pumping lemma, Applications
Closure Properties of Context-Free Languages: Closure properties of CFL’s, Decision Properties of CFL’s

Turing Machines: Introduction to Turing Machine, Formal Description, Instantaneous description, The language of a Turing machine

UNIT - V
Types of Turing machine: Turing machines and halting
Undecidability: Undecidability, A Language that is Not Recursively Enumerable, An Undecidable Problem That is RE, Undecidable Problems about Turing Machines, Recursive languages, Properties of recursive languages, Post's Correspondence Problem, Modified Post Correspondence problem, Other Undecidable Problems, Counter machines.

Textbooks:


References:

1. Introduction to Languages and The Theory of Computation, John C Martin, TMH.
SOFTWARE ENGINEERING

III Year B.Tech. I-Semester

<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>

Objectives
1. The aim of the course is to provide an understanding of the working knowledge of the techniques for estimation, design, testing and quality management of large software development projects.
2. Topics include process models, software requirements, software design, software testing, software process/product metrics, risk management, quality management and UML diagrams.

Outcomes
1. Ability to translate end-user requirements into system and software requirements, using e.g. UML, and structure the requirements in a Software Requirements Document (SRD).
2. Identify and apply appropriate software architectures and patterns to carry out high level design of a system and be able to critically compare alternative choices.
3. Will have experience and/or awareness of testing problems and will be able to develop a simple testing report.

UNIT - I
Introduction to Software Engineering: The evolving role of software, changing nature of software, software myths.
A Generic view of process: Software engineering- a layered technology, a process framework, the capability maturity model integration (CMMI), process patterns, process assessment, personal and team process models.
Process models: The waterfall model, incremental process models, evolutionary process models, the unified process.

UNIT - II
Software Requirements: Functional and non-functional requirements, user requirements, system requirements, interface specification, the software requirements document.
Requirements engineering process: Feasibility studies, requirements elicitation and analysis, requirements validation, requirements management.
System models: Context models, behavioral models, data models, object models, structured methods.

UNIT - III
Design Engineering: Design process and design quality, design concepts, the design model.
Creating an architectural design: software architecture, data design, architectural styles and patterns, architectural design, conceptual model of UML, basic structural modeling, class diagrams, sequence diagrams, collaboration diagrams, use case diagrams, component diagrams.

UNIT - IV
Testing Strategies: A strategic approach to software testing, test strategies for conventional software, black-box and white-box testing, validation testing, system testing, the art of debugging.
Product metrics: Software quality, metrics for analysis model, metrics for design model, metrics for source code, metrics for testing, metrics for maintenance.

UNIT - V
Metrics for Process and Products: Software measurement, metrics for software quality.
Risk management: Reactive Vs proactive risk strategies, software risks, risk identification, risk projection, risk refinement, RMMM, RMMM plan.
Quality Management: Quality concepts, software quality assurance, software reviews, formal technical reviews, statistical software quality assurance, software reliability, the ISO 9000 quality standards.
Textbooks:
3. The Unified Modeling Language user guide Grady Booch, James Rumbaugh, Ivar Jacobson, Pearson Education.

References:
COMPUTER NETWORKS

III Year B.Tech. I-Semester

<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>

Prerequisites
1. A course on “Programming for problem solving”
2. A course on “Data Structures”

Objectives
1. The objective of the course is to equip the students with a general overview of the concepts and fundamentals of computer networks.
2. Familiarize the students with the standard models for the layered approach to communication between machines in a network and the protocols of the various layers.

Outcomes
1. Gain the knowledge of the basic computer network technology.
2. Gain the knowledge of the functions of each layer in the OSI and TCP/IP reference model.
3. Obtain the skills of subnetting and routing mechanisms.
4. Familiarity with the essential protocols of computer networks, and how they can be applied in network design and implementation.

UNIT - I
Network hardware, Network software, OSI, TCP/IP Reference models, Example Networks: ARPANET, Internet.


UNIT - II
Data link layer: Design issues, framing, Error detection and correction.

Elementary data link protocols: simplex protocol, A simplex stop and wait protocol for an error-free channel, A simplex stop and wait protocol for noisy channel.

Sliding Window protocols: A one-bit sliding window protocol, A protocol using Go-Back-N, A protocol using Selective Repeat, Example data link protocols.

Medium Access sub layer: The channel allocation problem, Multiple access protocols: ALOHA, Carrier sense multiple access protocols, collision free protocols. Wireless LANs, Data link layer switching.

UNIT - III

UNIT - IV
Transport Layer: Transport Services, Elements of Transport protocols, Connection management, TCP and UDP protocols.

UNIT - V
Application Layer – Domain name system, SNMP, Electronic Mail; the World WEB, HTTP, Streaming audio and video.
Textbook:


References:

2. Data Communications and Networking – Behrouz A. Forouzan. 3rd Edition, TMH.
ARTIFICIAL INTELLIGENCE

III Year B.Tech. I-Semester

<table>
<thead>
<tr>
<th>Prerequisites</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. A course on “Computer Programming and Data Structures”</td>
</tr>
<tr>
<td>2. A course on “Advanced Data Structures”</td>
</tr>
<tr>
<td>3. A course on “Design and Analysis of Algorithms”</td>
</tr>
<tr>
<td>4. A course on “Mathematical Foundations of Computer Science”</td>
</tr>
<tr>
<td>5. Some background in linear algebra, data structures and algorithms, and probability will all be helpful</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Objectives</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. To learn the distinction between optimal reasoning Vs. human like reasoning</td>
</tr>
<tr>
<td>2. To understand the concepts of state space representation, exhaustive search, heuristic search together with the time and space complexities.</td>
</tr>
<tr>
<td>3. To learn different knowledge representation techniques.</td>
</tr>
<tr>
<td>4. To understand the applications of AI, namely game playing, theorem proving, and machine learning.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Outcomes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Ability to formulate an efficient problem space for a problem expressed in natural language.</td>
</tr>
<tr>
<td>2. Select a search algorithm for a problem and estimate its time and space complexities.</td>
</tr>
<tr>
<td>3. Possess the skill for representing knowledge using the appropriate technique for a given problem.</td>
</tr>
<tr>
<td>4. Possess the ability to apply AI techniques to solve problems of game playing, and machine learning.</td>
</tr>
</tbody>
</table>

UNIT - I

Problem Solving by Search-I

Introduction to AI, Intelligent Agents

Problem Solving by Search-II:

UNIT - II

Problem Solving by Search-II and Propositional Logic

Adversarial Search:
Games, Optimal Decisions in Games, Alpha–Beta Pruning, Imperfect Real-Time Decisions.

Constraint Satisfaction Problems:
Defining Constraint Satisfaction Problems, Constraint Propagation, Backtracking Search for CSPs, Local Search for CSPs, The Structure of Problems.
Propositional Logic:
Knowledge-Based Agents, The Wumpus World, Logic, Propositional Logic, Propositional Theorem
Proving: Inference and proofs, Proof by resolution, Horn clauses and definite clauses, Forward and backward chaining, Effective Propositional Model Checking, Agents Based on Propositional Logic.

UNIT - III
Logic and Knowledge Representation

First-Order Logic:
Representation, Syntax and Semantics of First-Order Logic, Using First-Order Logic, Knowledge Engineering in First-Order Logic.

Inference in First-Order Logic:
Propositional vs. First-Order Inference, Unification and Lifting, Forward Chaining, Backward Chaining, Resolution.

Knowledge Representation:
Ontological Engineering, Categories and Objects, Events, Mental Events and Mental Objects, Reasoning Systems for Categories, Reasoning with Default Information.

UNIT - IV
Planning

Classical Planning:

Planning and Acting in the Real World:
Time, Schedules, and Resources, Hierarchical Planning, Planning and Acting in Nondeterministic Domains, Multi agent Planning.

UNIT - V
Uncertain knowledge and Learning

Uncertainty:
Acting under Uncertainty, Basic Probability Notation, Inference Using Full Joint Distributions, Independence, Bayes’ Rule and Its Use,

Probabilistic Reasoning:
Representing Knowledge in an Uncertain Domain, The Semantics of Bayesian Networks, Efficient Representation of Conditional Distributions, Approximate Inference in Bayesian Networks, Relational and First-Order Probability, Other Approaches to Uncertain Reasoning; Dempster-Shafer theory.

Learning:
Forms of Learning, Supervised Learning, Learning Decision Trees, Knowledge in Learning: Logical Formulation of Learning, Knowledge in Learning, Explanation-Based Learning, Learning Using Relevance Information, Inductive Logic Programming.
Textbooks:


References:

3. Artificial Intelligence, Shivani Goel, Pearson Education.
4. Artificial Intelligence and Expert systems – Patterson, Pearson Education.
INFORMATION THEORY & CODING
(Professional Elective - I)

III Year B.Tech. I-Semester

Prerequisite
1. A Course on “Digital Communications”

Objectives:
1. To acquire the knowledge in measurement of information and errors.
2. Understand the importance of various codes for communication systems.
3. To design encoder and decoder of various codes.
4. To known the applicability of source and channel codes.

Outcomes:
1. Learn measurement of information and errors.
2. Obtain knowledge in designing various source codes and channel codes.
3. Design encoders and decoders for block and cyclic codes.
4. Understand the significance of codes in various applications.

UNIT - I
Coding for Reliable Digital Transmission and storage
Mathematical model of Information, A Logarithmic Measure of Information, Average and Mutual Information and Entropy, Types of Errors, Error Control Strategies.

Source Codes: Shannon-fano coding, Huffman coding

UNIT - II
Linear Block Codes: Introduction to Linear Block Codes, Syndrome and Error Detection, Minimum Distance of a Block code, Error-Detecting and Error-correcting Capabilities of a Block code, Standard array and Syndrome Decoding, Probability of an undetected error for Linear Codes over a BSC, Hamming Codes. Applications of Block codes for Error control in data storage system.

UNIT - III
Cyclic Codes: Description, Generator and Parity-check Matrices, Encoding, Syndrome Computation and Error Detection, Decoding, Cyclic Hamming Codes, Shortened cyclic codes, Error-trapping decoding for cyclic codes, Majority logic decoding for cyclic codes.

UNIT - IV
Convolutional Codes: Encoding of Convolutional Codes- Structural and Distance Properties, state, tree, trellis diagrams, maximum likelihood decoding, Sequential decoding, Majority- logic decoding of Convolution codes. Application of Viterbi Decoding and Sequential Decoding, Applications of Convolutional codes in ARQ system.

UNIT - V
BCH Codes: Minimum distance and BCH bounds, Decoding procedure for BCH codes, Syndrome computation and iterative algorithms, Error locations polynomials for single and double error correction.

Textbooks:
References:

2. Introduction to Error Control Codes, Salvatore Gravano, Oxford
3. Error Correction Coding – Mathematical Methods and Algorithms, Todd K. Moon, 2006, Wiley India.
ADVANCED COMPUTER ARCHITECTURE
(Professional Elective - I)

III Year B.Tech. I-Semester

Prerequisites: A Course on “Computer Organization”

Objectives
1. To impart the concepts and principles of parallel and advanced computer architectures.
2. To develop the design techniques of Scalable and multithreaded Architectures.
3. To Apply the concepts and techniques of parallel and advanced computer architectures to design
   modern computer systems

Outcomes
Gain knowledge of
1. Computational models and Computer Architectures.
2. Concepts of parallel computer models.
3. Scalable Architectures, Pipelining, Superscalar processors, multiprocessors

UNIT - I
Theory of Parallelism, Parallel computer models, The State of Computing, Multiprocessors and
Multicomputers, Multivector and SIMD Computers, PRAM and VLSI models, Architectural development
tracks, Program and network properties, Conditions of parallelism, Program partitioning and Scheduling,
Program flow Mechanisms, System interconnect Architectures.

UNIT - II
Principals of Scalable performance, Performance metrics and measures, Parallel Processing applications,
Speed up performance laws, Scalability Analysis and Approaches, Hardware Technologies, Processes and
Memory Hierarchy, Advanced Processor Technology, Superscalar and Vector Processors, Memory
Hierarchy Technology, Virtual Memory Technology.

UNIT - III
Bus Cache and Shared memory, Backplane bus systems, Cache Memory organizations, Shared-Memory
Organizations, Sequential and weak consistency models, Pipelining and superscalar techniques, Linear
Pipeline Processors, Non-Linear Pipeline Processors, Instruction Pipeline design, Arithmetic pipeline
design, superscalar pipeline design.

UNIT - IV
Parallel and Scalable Architectures, Multiprocessors and Multicomputers, Multiprocessor system
interconnects, cache coherence and synchronization mechanism, Three Generations of Multicomputers,
Message-passing Mechanisms, Multivector and SIMD computers, Vector Processing Principals, Multivector
Multiprocessors, Compound Vector processing, SIMD computer Organizations, The connection machine
CM-5,
UNIT - V
Scalable, Multithreaded and Dataflow Architectures, Latency-hiding techniques, Principals of Multithreading, Fine-Grain Multicomputers, Scalable and multithreaded Architectures, Dataflow and hybrid Architectures.

Textbook:

References:
DATA MINING
(Professional Elective - I)

III Year B.Tech. I-Semester

<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>

Prerequisites
1. A course on “Database Management Systems”
2. Knowledge of probability and statistics

Objectives
1. It presents methods for mining frequent patterns, associations, and correlations.
2. It then describes methods for data classification and prediction, and data–clustering approaches.
3. It covers mining various types of data stores such as spatial, textual, multimedia, streams.

Outcomes
1. Ability to understand the types of the data to be mined and present a general classification of tasks and primitives to integrate a data mining system.
2. Apply preprocessing methods for any given raw data.
3. Extract interesting patterns from large amounts of data.
4. Discover the role played by data mining in various fields.
5. Choose and employ suitable data mining algorithms to build analytical applications.
6. Evaluate the accuracy of supervised and unsupervised models and algorithms.

UNIT - I
Data Mining
Data–Types of Data–, Data Mining Functionalities– Interestingness Patterns–Classification of Data Mining systems– Data mining Task primitives –Integration of Data mining system with a Data warehouse–Major issues in Data Mining–Data Preprocessing.

UNIT – II
Association Rule Mining
Mining Frequent Patterns–Associations and correlations– Mining Methods– Mining Various kinds of Association Rules– Correlation Analysis– Constraint based Association mining. Graph Pattern Mining, SPM.

UNIT – III
Classification
UNIT – IV
Clustering And Applications
Cluster analysis—Types of Data in Cluster Analysis—Categorization of Major Clustering Methods—Partitioning Methods—Hierarchical Methods—Density-Based Methods—Grid-Based Methods, Outlier Analysis.

UNIT - V
Advanced Concepts

Textbooks:
1. Data Mining – Concepts and Techniques – Jiawei Han & Micheline Kamber, 3rd Edition Elsevier.
2. Data Mining Introductory and Advanced topics –Margaret H Dunham, PEA.

References:
III Year B.Tech. I-Semester

Prerequisites
1. Students are expected to have knowledge in linear signals and systems, Fourier Transform, basic linear algebra, basic probability theory and basic programming techniques; knowledge of Digital Signal Processing is desirable.
2. A course on “Computational Mathematics”
3. A course on “Computer Oriented Statistical Methods”

Objectives
1. Provide a theoretical and mathematical foundation of fundamental Digital Image Processing concepts.
2. The topics include image acquisition; sampling and quantization; preprocessing; enhancement; restoration; segmentation; and compression.

Outcomes
1. Demonstrate the knowledge of the basic concepts of two-dimensional signal acquisition, sampling, and quantization.
2. Demonstrate the knowledge of filtering techniques.
3. Demonstrate the knowledge of 2D transformation techniques.
4. Demonstrate the knowledge of image enhancement, segmentation, restoration and compression techniques.

UNIT - I

UNIT - II

UNIT - III

UNIT - IV
Image Segmentation Detection of Discontinuities, Edge Linking and Boundary Detection, Thresholding, Region Oriented Segmentation.

UNIT - V
Textbook:


References:

PRINCIPLES OF PROGRAMMING LANGUAGES
(Professional Elective-I)

III Year B.Tech. CSE I-Semester

<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>

Prerequisites
1. A course on “Mathematical Foundations of Computer Science”
2. A course on “Computer Programming and Data Structures”

Objectives
1. Introduce important paradigms of programming languages
2. To provide conceptual understanding of high level language design and implementation
3. Topics include programming paradigms; syntax and semantics; data types, expressions and statements; subprograms and blocks; abstract data types; concurrency; functional and logic programming languages; and scripting languages

Outcomes
1. Acquire the skills for expressing syntax and semantics in formal notation
2. Identify and apply a suitable programming paradigm for a given computing application
3. Gain knowledge of and able to compare the features of various programming languages

UNIT - I

Syntax and Semantics: General Problem of Describing Syntax and Semantics, Formal Methods of Describing the Meanings of Programs

UNIT - II
Names, Bindings, and Scopes: Introduction, Names, Variables, Concept of Binding, Scope, Scope and Lifetime, Referencing Environments, Named Constants

Data Types: Introduction, Primitive Data Types, Character String Types, User Defined Ordinal Types, Array, Associative Arrays, Record, Union, Tuple Types, List Types, Pointer and Reference Types, Type Checking, Strong Typing, Type Equivalence

Expressions and Statements: Arithmetic Expressions, Overloaded Operators, Type Conversions, Relational and Boolean Expressions, Short Circuit Evaluation, Assignment Statements, Mixed-Mode Assignment


UNIT - III
Subprograms and Blocks: Fundamentals of Sub-Programs, Design Issues for Subprograms, Local Referencing Environments, Parameter Passing Methods, Parameters that Are Subprograms, Calling Subprograms Indirectly, Overloaded Subprograms, Generic Subprograms, Design Issues for Functions, User Defined Overloaded Operators, Closures, Coroutines
Implementing Subprograms: General Semantics of Calls and Returns, Implementing Simple Subprograms, Implementing Subprograms with Stack-Dynamic Local Variables, Nested Subprograms, Blocks, Implementing Dynamic Scoping

Abstract Data Types: The Concept of Abstraction, Introductions to Data Abstraction, Design Issues, Language Examples, Parameterized ADT, Encapsulation Constructs, Naming Encapsulations

UNIT - IV
Concurrency: Introduction, Introduction to Subprogram Level Concurrency, Semaphores, Monitors, Message Passing, Java Threads, Concurrency in Function Languages, Statement Level Concurrency.

Exception Handling and Event Handling: Introduction, Exception Handling in Ada, C++, Java, Introduction to Event Handling, Event Handling with Java and C#.

UNIT - V
Functional Programming Languages: Introduction, Mathematical Functions, Fundamentals of Functional Programming Language, LISP, Support for Functional Programming in Primarily Imperative Languages, Comparison of Functional and Imperative Languages

Logic Programming Language: Introduction, an Overview of Logic Programming, Basic Elements of Prolog, Applications of Logic Programming.

Scripting Language: Pragmatics, Key Concepts, Case Study: Python – Values and Types, Variables, Storage and Control, Bindings and Scope, Procedural Abstraction, Data Abstraction, Separate Compilation, Module Library. (Text Book 2)

Textbooks:

References:
COMPUTER GRAPHICS
(Professional Elective - II)

III Year B.Tech. CSE I-Semester

Prerequisites
1. Familiarity with the theory and use of coordinate geometry and of linear algebra such as matrix multiplication.
2. A course on “Computer Programming and Data Structures”

Objectives
1. The aim of this course is to provide an introduction of fundamental concepts and theory of computer graphics.
2. Topics covered include graphics systems and input devices; geometric representations and 2D/3D transformations; viewing and projections; illumination and color models; animation; rendering and implementation; visible surface detection;

Outcomes
1. Acquire familiarity with the relevant mathematics of computer graphics.
2. Be able to design basic graphics application programs, including animation.
3. Be able to design applications that display graphic images to given specifications.

UNIT - I
Introduction: Application areas of Computer Graphics, overview of graphics systems, video-display devices, raster-scan systems, random scan systems, graphics monitors and work stations and input devices.

Output primitives: Points and lines, line drawing algorithms (Bresenham’s and DDA Algorithm), mid-point circle and ellipse algorithms.

Polygon Filling: Scan-line algorithm, boundary-fill and flood-fill algorithms.

UNIT - II
2-D geometrical transforms: Translation, scaling, rotation, reflection and shear transformations, matrix representations and homogeneous coordinates, composite transforms, transformations between coordinate systems.

2-D viewing: The viewing pipeline, viewing coordinate reference frame, window to view-port coordinate transformation, viewing functions, Cohen-Sutherland algorithms, Sutherland-Hodgeman polygon clipping algorithm.

UNIT - III
3-D object representation: Polygon surfaces, quadric surfaces, spline representation, Hermite curve, Bezier curve and B-Spline curves, Bezier and B-Spline surfaces. Basic illumination models, polygon rendering methods.
UNIT - IV
3-D Geometric transformations: Translation, rotation, scaling, reflection and shear transformations, composite transformations.

3-D viewing: Viewing pipeline, viewing coordinates, view volume and general projection transforms and clipping.

UNIT - V
Computer animation: Design of animation sequence, general computer animation functions, raster animation, computer animation languages, key frame systems, motion specifications

Visible surface detection methods: Classification, back-face detection, depth-buffer, BSP-tree methods and area sub-division methods

Textbooks:

3. Computer Graphics, Steven Harrington, TMH

References:

ADVANCED OPERATING SYSTEMS
(Professional Elective - II)

III Year B.Tech. CSE I-Semester

Objectives
1. To study, learn, and understand the main concepts of advanced operating systems (parallel processing systems, distributed systems, real time systems, network operating systems, and open source operating systems).
2. Hardware and software features that support these systems.

Outcomes
1. Understand the design approaches of advanced operating systems.
2. Analyze the design issues of distributed operating systems.
3. Evaluate design issues of multi processor operating systems.
4. Identify the requirements Distributed File System and Distributed Shared Memory.
5. Formulate the solutions to schedule the real time applications.

UNIT - I

UNIT - II

UNIT - III

UNIT - IV

UNIT - V

UNIT - V
**Distributed Shared Memory**: Architecture and Motivation, Algorithms for Implementing DSM, Memory Coherence, Coherence Protocols, Design Issues

**Textbook:**


**Reference:**

INFORMATION RETRIEVAL SYSTEMS
(Professional Elective - II)

III Year B.Tech. CSE I-Semester

Prerequisites
1. A Course on “Data Structures”

Objectives
1. To learn the important concepts and algorithms in IRS
2. To understand the data/file structures that are necessary to design, and implement information retrieval (IR) systems.

Outcomes
1. Ability to apply IR principles to locate relevant information large collections of data
2. Ability to design different document clustering algorithms
3. Implement retrieval systems for web search tasks.

UNIT - I

Information Retrieval System Capabilities: Search Capabilities, Browse Capabilities, Miscellaneous Capabilities

UNIT - II
Cataloging and Indexing: History and Objectives of Indexing, Indexing Process, Automatic Indexing, Information Extraction


UNIT - III
Automatic Indexing: Classes of Automatic Indexing, Statistical Indexing, Natural Language, Concept Indexing, Hypertext Linkages

Document and Term Clustering: Introduction to Clustering, Thesaurus Generation, Item Clustering, Hierarchy of Clusters

UNIT - IV
User Search Techniques: Search Statements and Binding, Similarity Measures and Ranking, Relevance Feedback, Selective Dissemination of Information Search, Weighted Searches of Boolean Systems, Searching the INTERNET and Hypertext

Information Visualization: Introduction to Information Visualization, Cognition and Perception, Information Visualization Technologies
UNIT - V
Text Search Algorithms: Introduction to Text Search Techniques, Software Text Search Algorithms, Hardware Text Search Systems

Multimedia Information Retrieval: Spoken Language Audio Retrieval, Non-Speech Audio Retrieval, Graph Retrieval, Imagery Retrieval, Video Retrieval

Textbook:

References:
2. Information Storage & Retrieval, Robert Korfhage, John Wiley & Sons.
ADVANCED DATABASES
(Professional Elective - II)

III Year B.Tech. CSE I-Semester  

<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>

Prerequisites
1. A course on “Database Management Systems”

Objectives
1. The purpose of the course is to enrich the previous knowledge of database systems and exposing the need for distributed database technology to confront with the deficiencies of the centralized database systems.
2. Introduce basic principles and implementation techniques of distributed database systems.
3. Equip students with principles and knowledge of parallel and object oriented databases.
4. Topics include distributed DBMS architecture and design; query processing and optimization; distributed transaction management and reliability; parallel and object database management systems.

Outcomes
1. Understand theoretical and practical aspects of distributed database systems.
2. Study and identify various issues related to the development of distributed database system.
3. Understand the design aspects of object oriented database system and related development.

UNIT - I
Introduction: Distributed Data Processing, Distributed Database System, Promises of DDBSs, Problem areas.


UNIT - II
Query processing and decomposition: Query processing objectives, characterization of query processors, layers of query processing, query decomposition, localization of distributed data.

Distributed query Optimization: Query optimization, centralized query optimization, distributed query optimization algorithms.

UNIT - III
Transaction Management: Definition, properties of transaction, types of transactions, distributed concurrency control: serializability, concurrency control mechanisms & algorithms, time - stamped & optimistic concurrency control Algorithms, deadlock Management.

UNIT - IV
Distributed DBMS Reliability: Reliability concepts and measures, fault-tolerance in distributed systems, failures in Distributed DBMS, local & distributed reliability protocols, site failures and network partitioning.
**Parallel Database Systems**: Parallel database system architectures, parallel data placement, parallel query processing, load balancing, database clusters.

**UNIT - V**

**Distributed object Database Management Systems**: Fundamental object concepts and models, object distributed design, architectural issues, object management, distributed object storage, object query Processing.

**Object Oriented Data Model**: Inheritance, object identity, persistent programming languages, persistence of objects, comparison OODBMS and ORDBMS

**Textbooks:**

2. Distributed Databases, Stefano Ceri and Giuseppe Pelagatti, McGraw Hill.

**References:**

NATURAL LANGUAGE PROCESSING
(Professional Elective - II)

III Year B.Tech. CSE I-Semester

L  T  P  C
3  0  0  3

Prerequisites
1. A Course on “Data structures”
2. A Course on “Finite automata and Probability theory”

Objectives
1. Introduce to some of the problems and solutions of NLP and their relation to linguistics and statistics.

Outcomes
1. Show sensitivity to linguistic phenomena and an ability to model them with formal grammars.
2. Understand and carry out proper experimental methodology for training and evaluating empirical NLP systems
3. Able to manipulate probabilities, construct statistical models over strings and trees, and estimate parameters using supervised and unsupervised training methods.
4. Able to design, implement, and analyze NLP algorithms
5. Able to design different language modeling Techniques.

UNIT - I
Finding the Structure of Words: Words and Their Components, Issues and Challenges, Morphological Models
Finding the Structure of Documents: Introduction, Methods, Complexity of the Approaches, Performances of the Approaches

UNIT - II

UNIT - III

UNIT - IV
Predicate-Argument Structure, Meaning Representation Systems, Software

UNIT - V
Discourse Processing: Cohesion, Reference Resolution, Discourse Cohesion and Structure

Textbooks
1. Multilingual natural Language Processing Applications: From Theory to Practice, Daniel M. Bikel and Imed Zitouni, Pearson Publication
2. Natural Language Processing and Information Retrieval, Tanvier Siddiqui, U.S. Tiwary

Reference
1. Speech and Natural Language Processing, Daniel Jurafsky & James H Martin, Pearson Publications
SOFTWARE ENGINEERING LAB

III Year B.Tech. CSE I-Semester

<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1.5</td>
</tr>
</tbody>
</table>

**Prerequisites**
1. A course on “Programming for Problem Solving”

**Co-requisite**
1. A Course on “Software Engineering”

**Objectives**
1. To have hands on experience in developing a software project by using various software engineering principles and methods in each of the phases of software development.

**Outcomes**
1. Ability to translate end-user requirements into system and software requirements
2. Ability to generate a high level design of the system from the software requirements
3. Will have experience and/or awareness of testing problems and will be able to develop a simple testing report

**List of Experiments**

Do the following 8 exercises for any two projects given in the list of sample projects or any other projects:

4. Study and usage of any Design phase CASE tool
5. Performing the Design by using any Design phase CASE tools.
6. Develop test cases for unit testing and integration testing
7. Develop test cases for various white box and black box testing techniques.

**Sample Projects:**
1. Passport automation System
2. Book Bank
3. Online Exam Registration
4. Stock Maintenance System
5. Online course reservation system
6. E-ticketing
7. Software Personnel Management System
8. Credit Card Processing
10. Recruitment system

**Textbooks:**
3. The unified modeling language user guide Grady Booch, James Rambaugh, Ivar Jacobson, Pearson Education.
COMPUTER NETWORKS LAB

III Year B.Tech. CSE I-Semester

<table>
<thead>
<tr>
<th></th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1.5</td>
</tr>
</tbody>
</table>

Objectives
1. To understand the working principle of various communication protocols.
2. To understand the network simulator environment and visualize a network topology and observe its performance.
3. To analyze the traffic flow and the contents of protocol frames.

Outcomes
1. Implement data link layer farming methods.
2. Analyze error detection and error correction codes.
3. Implement and analyze routing and congestion issues in network design.
4. Implement Encoding and Decoding techniques used in presentation layer.
5. To be able to work with different network tools.

List of Experiments
1. Implement the data link layer framing methods such as character, character-stuffing and bit stuffing.
2. Write a program to compute CRC code for the polynomials CRC-12, CRC-16 and CRC CCIP.
3. Develop a simple data link layer that performs the flow control using the sliding window protocol, and loss recovery using the Go-Back-N mechanism.
4. Implement Dijkstra’s algorithm to compute the shortest path through a network.
5. Take an example subnet of hosts and obtain a broadcast tree for the subnet.
6. Implement distance vector routing algorithm for obtaining routing tables at each node.
7. Implement data encryption and data decryption.
8. Write a program for congestion control using Leaky bucket algorithm.
9. Write a program for frame sorting technique used in buffers.
10. **Wireshark**
    i. Packet Capture Using Wire shark
    ii. Starting Wire shark
    iii. Viewing Captured Traffic
11. How to run Nmap scan
12. Operating System Detection using Nmap
13. Do the following using NS2 Simulator
    i. NS2 Simulator-Introduction
    ii. Simulate to Find the Number of Packets Dropped
    iii. Simulate to Find the Number of Packets Dropped by TCP/UDP
    iv. Simulate to Find the Number of Packets Dropped due to Congestion
    v. Simulate to Compare Data Rate& Throughput.
    vi. Simulate to Plot Congestion for Different Source/Destination
    vii. Simulate to Determine the Performance with respect to Transmission of Packets

Textbook:

References:
2. Data Communications and Networking – Behrouz A. Forouzan. 3rd Edition, TMH.
ADVANCED ENGLISH COMMUNICATION SKILLS (AECS) LAB

III Year B.Tech. CSE I-Semester

<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

1. **Introduction**

The introduction of the Advanced Communication Skills Lab is considered essential at 3rd year level. At this stage, the students need to prepare themselves for their careers which may require them to listen to, read, speak and write in English both for their professional and interpersonal communication in the globalised context.

The proposed course should be a laboratory course to enable students to use ‘good’ English and perform the following:

- Gathering ideas and information to organize ideas relevantly and coherently.
- Engaging in debates.
- Participating in group discussions.
- Facing interviews.
- Writing project/research reports/technical reports.
- Making oral presentations.
- Writing formal letters.
- Transferring information from non-verbal to verbal texts and vice-versa.
- Taking part in social and professional communication.

2. **Objectives:**

This Lab focuses on using multi-media instruction for language development to meet the following targets:

1. To improve the students’ fluency in English, through a well-developed vocabulary and enable them to listen to English spoken at normal conversational speed by educated English speakers and respond appropriately in different socio-cultural and professional contexts.
2. Further, they would be required to communicate their ideas relevantly and coherently in writing.
3. To prepare all the students for their placements.

3. **Syllabus:**

The following course content to conduct the activities is prescribed for the Advanced English Communication Skills (AECS) Lab:

1. **Activities on Fundamentals of Inter-personal Communication and Building Vocabulary** - Starting a conversation – responding appropriately and relevantly – using the right body language – Role Play in different situations & Discourse Skills- using visuals - Synonyms and antonyms, word roots, one-word substitutes, prefixes and suffixes, study of word origin, business vocabulary, analogy, idioms and phrases, collocations & usage of vocabulary.
2. **Activities on Reading Comprehension** – General Vs Local comprehension, reading for facts, guessing meanings from context, scanning, skimming, inferring meaning, critical reading & effective googling.

4. **Activities on Presentation Skills** – Oral presentations (individual and group) through JAM sessions/seminars/PPTs and written presentations through posters/projects/reports/emails/assignments etc.

5. **Activities on Group Discussion and Interview Skills** – Dynamics of group discussion, intervention, summarizing, modulation of voice, body language, relevance, fluency and organization of ideas and rubrics for evaluation- Concept and process, pre-interview planning, opening strategies, answering strategies, interview through tele-conference & video-conference and Mock Interviews.

4. **Minimum Requirement:**
The Advanced English Communication Skills (AECS) Laboratory shall have the following infrastructural facilities to accommodate at least 35 students in the lab:

- Spacious room with appropriate acoustics.
- Round Tables with movable chairs
- Audio-visual aids
- LCD Projector
- Public Address system
- T. V, a digital stereo & Camcorder
- Headphones of High quality

5. **Suggested Software:**
The software consisting of the prescribed topics elaborated above should be procured and used.

- [Oxford Advanced Learner's Compass, 7th Edition](#)
- [DELTA’s key to the Next Generation TOEFL Test: Advanced Skill Practice](#)
- [Lingua TOEFL CBT Insider](#) by Dreamtech
- [TOEFL & GRE](#) (KAPLAN, AARCO&BARRONS, USA, Cracking GRE by CLIFFS)

6. **Books Recommended:**
CONSTITUTION OF INDIA
(Mandate Course)

III Year B.Tech. CSE I-Semester          L   T   P   C
                                      2   0   0   0

Objectives
1. To understand the premises informing the twin themes of liberty and freedom from a civil rights perspective.
2. To address the growth of Indian opinion regarding modern Indian intellectuals’ constitutional role and entitlement to civil and economic rights as well as the emergence of nationhood in the early years of Indian nationalism.
3. To address the role of socialism in India after the commencement of the Bolshevik Revolution in 1917 and its impact on the initial drafting of the Indian Constitution.

Outcomes
1. To discuss the growth of the demand for civil rights in India for the bulk of Indians before the arrival of Gandhi in Indian politics.
2. To discuss the intellectual origins of the framework of argument that informed the conceptualization of social reforms leading to revolution in India.
3. Discuss the circumstances surrounding the foundation of the Congress Socialist Party [CSP] under the leadership of Jawaharlal Nehru and the eventual failure of the proposal of direct elections through adult suffrage in the Indian Constitution.

UNIT – I
History of Making of the Indian Constitution: History, Drafting Committee, (Composition & Working)

UNIT – II
Philosophy of the Indian Constitution: Preamble, Salient Features

UNIT - III
Contours of Constitutional Rights & Duties:
   Fundamental Rights
   Right to Equality
   Right to Freedom
   Right against Exploitation
   Right to Freedom of Religion
   Cultural and Educational Rights
   Right to Constitutional Remedies
   Directive Principles of State Policy
   Fundamental Duties.

UNIT – IV
Organs of Governance:
   Parliament
   Composition
   Qualifications and Disqualifications
   Powers and Functions
   President
   Governor
   Council of Ministers
   Judiciary, Appointment and Transfer of Judges, Qualifications
   Powers and Functions
UNIT – V
Local Administration:
District’s Administration head: Role and Importance,
Municipalities: Introduction, Mayor and role of Elected Representative CEO of Municipal Corporation.
Elected officials and their roles, CEO ZilaPachayat: Position and role.
Block level: Organizational Hierarchy (Different departments),
Village level: Role of Elected and Appointed officials,
Importance of grass root democracy

UNIT – VI
Election Commission:
Election Commission: Role and Functioning.
Chief Election Commissioner and Election Commissioners.
State Election Commission: Role and Functioning.
Institute and Bodies for the welfare of SC/ST/OBC and women.

Textbooks:
1. The Constitution of India, 1950 (Bare Act), Government Publication.
MACHINE LEARNING

III Year B.Tech. CSE II-Semester

Prerequisites
1. A Course on “Data Structures”
2. Knowledge on statistical methods

Objectives
1. This course explains machine learning techniques such as decision tree learning, Bayesian learning etc.
2. To understand computational learning theory.
3. To study the pattern comparison techniques.

Outcomes
1. Understand the concepts of computational intelligence like machine learning
2. Ability to get the skill to apply machine learning techniques to address the real time problems in different areas
3. Understand the Neural Networks and its usage in machine learning application.

UNIT - I
Introduction - Well-posed learning problems, designing a learning system, Perspectives and issues in machine learning

Concept learning and the general to specific ordering – introduction, a concept learning task, concept learning as search, find-S: finding a maximally specific hypothesis, version spaces and the candidate elimination algorithm, remarks on version spaces and candidate elimination, inductive bias.

Decision Tree Learning – Introduction, decision tree representation, appropriate problems for decision tree learning, the basic decision tree learning algorithm, hypothesis space search in decision tree learning, inductive bias in decision tree learning, issues in decision tree learning.

UNIT - II
Artificial Neural Networks-1– Introduction, neural network representation, appropriate problems for neural network learning, perceptions, multilayer networks and the back-propagation algorithm.


Evaluation Hypotheses – Motivation, estimation hypothesis accuracy, basics of sampling theory, a general approach for deriving confidence intervals, difference in error of two hypotheses, comparing learning algorithms.

UNIT - III
Bayesian learning – Introduction, Bayes theorem, Bayes theorem and concept learning, Maximum Likelihood and least squared error hypotheses, maximum likelihood hypotheses for predicting probabilities, minimum description length principle, Bayes optimal classifier, Gibs algorithm, Naïve Bayes classifier, an example: learning to classify text, Bayesian belief networks, the EM algorithm.
Computational learning theory – Introduction, probably learning an approximately correct hypothesis, sample complexity for finite hypothesis space, sample complexity for infinite hypothesis spaces, the mistake bound model of learning.

Instance-Based Learning- Introduction, $k$-nearest neighbour algorithm, locally weighted regression, radial basis functions, case-based reasoning, remarks on lazy and eager learning.

UNIT- IV

Genetic Algorithms – Motivation, Genetic algorithms, an illustrative example, hypothesis space search, genetic programming, models of evolution and learning, parallelizing genetic algorithms.


Reinforcement Learning – Introduction, the learning task, $Q$-learning, non-deterministic, rewards and actions, temporal difference learning, generalizing from examples, relationship to dynamic programming.

UNIT - V

Analytical Learning-1- Introduction, learning with perfect domain theories: PROLOG-EBG, remarks on explanation-based learning, explanation-based learning of search control knowledge.

Analytical Learning-2- Using prior knowledge to alter the search objective, using prior knowledge to augment search operators.

Combining Inductive and Analytical Learning – Motivation, inductive-analytical approaches to learning, using prior knowledge to initialize the hypothesis.

Textbook:

References:
COMPILER DESIGN

III Year B.Tech. CSE II-Semester

Prerequisites

1. A course on “Formal Languages and Automata Theory”
2. A course on “Computer Organization and architecture”
3. A course on “Computer Programming and Data Structures”

Objectives

1. Introduce the major concepts of language translation and compiler design and impart the knowledge of practical skills necessary for constructing a compiler.
2. Topics include phases of compiler, parsing, syntax directed translation, type checking use of symbol tables, code optimization techniques, intermediate code generation, code generation and data flow analysis.

Outcomes

1. Demonstrate the ability to design a compiler given a set of language features.
2. Demonstrate the the knowledge of patterns, tokens & regular expressions for lexical analysis.
3. Acquire skills in using lex tool & yacc tool for developing a scanner and parser.
4. Design and implement LL and LR parsers
5. Design algorithms to do code optimization in order to improve the performance of a program in terms of space and time complexity.
6. Design algorithms to generate machine code.

UNIT - I

Introduction: The structure of a compiler, the science of building a compiler, programming language basics

Lexical Analysis: The Role of the Lexical Analyzer, Input Buffering, Recognition of Tokens, The Lexical-Analyzer Generator Lex, Finite Automata, From Regular Expressions to Automata, Design of a Lexical-Analyzer Generator, Optimization of DFA-Based Pattern Matchers.

UNIT - II


UNIT - III

Syntax-Directed Translation: Syntax-Directed Definitions, Evaluation Orders for SDD's, Applications of Syntax-Directed Translation, Syntax-Directed Translation Schemes, Implementing L-Attributed SDD's.

UNIT - IV

**Run-Time Environments:** Stack Allocation of Space, Access to Nonlocal Data on the Stack, Heap Management, Introduction to Garbage Collection, Introduction to Trace-Based Collection.


UNIT - V


**Textbook:**


**References:**

1. lex & yacc – John R. Levine, Tony Mason, Doug Brown, O’reilly
WEB TECHNOLOGIES

III Year B.Tech. CSE II-Semester

Objectives

1. To introduce PHP language for server side scripting
2. To introduce XML and processing of XML Data with Java
3. To introduce Server side programming with Java Servlets and JSP
4. To introduce Client side scripting with Javascript and AJAX.

Outcomes

1. Gain knowledge of client side scripting, validation of forms and AJAX programming
2. Have understanding of server side scripting with PHP language
3. Have understanding of what is XML and how to parse and use XML Data with Java
4. To introduce Server side programming with Java Servlets and JSP

UNIT-I

Introduction to PHP: Declaring variables, data types, arrays, strings, operators, expressions, control structures, functions, Reading data from web form controls like text boxes, radio buttons, lists etc., Handling File Uploads. Connecting to database (MySQL as reference), executing simple queries, handling results, Handling sessions and cookies

File Handling in PHP: File operations like opening, closing, reading, writing, appending, deleting etc. on text and binary files, listing directories.

UNIT-II

HTML Common tags- List, Tables, images, forms, Frames; Cascading Style sheets;


UNIT - III

Introduction to Servlets: Common Gateway Interface (CGt), Life cycle of a Servlet, deploying a servlet, The Servlet API, Reading Servlet parameters, Reading Initialization parameters, Handling Http Request & Responses, Using Cookies and Sessions, connecting to a database using JDBC.

UNIT - IV

Introduction to JSP: The Anatomy of a JSP Page, JSP Processing, Declarations, Directives, Expressions, Code Snippets, implicit objects, Using Beans in JSP Pages, Using Cookies and session for session tracking, connecting to database in JSP.

UNIT - V

Client side Scripting: Introduction to Javascript, Javascript language – declaring variables, scope of variables, functions. event handlers (onclick, onsubmit etc.), Document Object Model, Form validation.
Textbooks:
1. Web Technologies, Uttam K Roy, Oxford University Press
2. The Complete Reference PHP — Steven Holzner, Tata McGraw-Hill

References:
2. Java Server Pages, Hans Bergsten, SPD O’Reilly,
4. Beginning Web Programming-Jon Duckett WROX.
6. Internet and World Wide Web — How to program, Dietel and Nieto, Pearson.
CONCURRENT PROGRAMMING
(Professional Elective - III)

III Year B.Tech. CSE II-Semester

<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>

Prerequisites

1. A course on “Operating Systems”
2. A course on “Java Programming”

Objectives

1. To explore the abstractions used in concurrent programming

Outcomes

1. Ability to implement the mechanisms for communication and co-ordination among concurrent processes.
2. Ability to understand and reason about concurrency and concurrent objects
3. Ability to implement the locking and non-blocking mechanisms
4. Ability to understand concurrent objects

UNIT - I

UNIT - II
Concurrent Objects - Concurrency and Correctness, Sequential Objects, Quiescent consistency, Sequential Consistency, Linearizability, Linearization Points, Formal Definitions

UNIT - III

UNIT - IV
Linked Lists: The Role of Locking, Introduction, List-Based Sets, Concurrent Reasoning, Coarse-Grained Synchronization, Fine-Grained Synchronization, Optimistic Synchronization, Lazy Synchronization, Non-Blocking Synchronization

UNIT - V
Concurrent Queues and the ABA Problem, Concurrent Stacks and Elimination, Transactional Memories
Textbook:


References:


NETWORK PROGRAMMING  
(Professional Elective - III)

III Year B.Tech. CSE II-Semester  
<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>

Objectives  
1. To understand inter process and inter-system communication  
2. To understand socket programming in its entirety  
3. To understand usage of TCP/UDP / Raw sockets  
4. To understand how to build network applications

Outcomes  
1. To write socket API based programs  
2. To design and implement client-server applications using TCP and UDP sockets  
3. To analyze network programs

UNIT - I  
Introduction to Network Programming: OSI model, Unix standards, TCP and UDP & TCP connection establishment and Format, Buffer sizes and limitation, standard internet services, Protocol usage by common internet application.

Sockets: Address structures, value – result arguments, Byte ordering and manipulation function and related functions Elementary TCP sockets: Socket, connect, bind, listen, accept, fork and exec function, concurrent servers. Close function and related function.

UNIT - II  
TCP client server: Introduction, TCP Echo server functions, Normal startup, terminate and signal handling server process termination, Crashing and Rebooting of server host shutdown of server host.

Elementary UDP sockets: Introduction UDP Echo server function, lost datagram, summary of UDP example, Lack of flow control with UDP, determining outgoing interface with UDP.

I/O Multiplexing: I/O Models, select function, Batch input, shutdown function, poll function, TCP Echo server,

UNIT - III  
Socket options: getsockopt and setsockopt functions. Socket states, Generic socket option IPV6 socket option ICMPV6 socket option IPV6 socket option and TCP socket options.


UNIT - IV  
Elementary name and Address conversions: DNS, gethostbyname function, Resolver option, Function and IPV6 support, uname function, other networking information.

Daemon Processes and inetd Superserver – Introduction, syslog daemon, syslog Function, daemon-init Function, inetd Daemon, daemon_inetd Function
**Broadcasting** - Introduction, Broadcast Addresses, Unicast versus Broadcast, dg_cli Function Using Broadcasting, Race Conditions

**Multicasting** - Introduction, Multicast Addresses, Multicasting versus Broadcasting on A LAN, Multicasting on a WAN, Multicast Socket Options, mcast_join and Related Functions, dg_cli Function Using Multicasting, Receiving MBone Session Announcements, Sending and Receiving, SNTP: Simple Network Time Protocol, SNTP (Continued)

**UNIT - V**
Raw Sockets-Introduction, Raw Socket Creation, Raw Socket Output, Raw Socket Input, Ping Program, Traceroute Program, An ICMP Message Daemon,
Remote Login: Terminal line disciplines, Pseudo-Terminals, Terminal modes, Control Terminals, rlogin Overview, RPC Transparency Issues.

**Textbooks:**

1. UNIX Network Programming, W. Richard Stevens, Bill Fenner, Andrew M. Rudoff, Pearson Education

**References:**

1. UNIX Systems Programming using C++ T CHAN, PHI.
2. UNIX for Programmers and Users, Graham GLASS, King abls, 3rd Edition, Pearson Education
SCRIPTING LANGUAGES
(Professional Elective - III)

III Year B.Tech. CSE II-Semester

L    T    P    C
3    0    0   3

Prerequisites
1. A course on “Computer Programming and Data Structures”
2. A course on “Object Oriented Programming Concepts”

Objectives
1. This course provides an introduction to the script programming paradigm
2. Introduces scripting languages such as Perl, Ruby and TCL.
3. Learning TCL

Outcomes
1. Comprehend the differences between typical scripting languages and typical system and application programming languages.
2. Gain knowledge of the strengths and weakness of Perl, TCL and Ruby; and select an appropriate language for solving a given problem.
3. Acquire programming skills in scripting language

UNIT - I
Introduction : Ruby ,Rails, The structure and Execution of Ruby Programs ,Package Management with RUBYGEMS, Ruby and web : Writing CGI scripts , cookies, Choice of Webservers ,SOAP and webservices

RubyTk – Simple Tk Application ,widgets , Binding events , Canvas ,scrolling

UNIT - II
Extending Ruby : Ruby Objects in C , the Jukebox extension, Memory allocation ,Ruby Type System , Embedding Ruby to Other Languages , Embedding a Ruby Interpreter

UNIT - III
Introduction to PERL and Scripting

Scripts and Programs, Origin of Scripting , Scripting Today, Characteristics of Scripting Languages,Uses for Scripting Languages, Web Scripting, and the universe of Scripting Languages. PERL- Names and Values, Variables, Scalar Expressions, Control Structures, arrays, list, hashes, strings, pattern and regular expressions, subroutines.

UNIT - IV
Advanced perl

Finer points of looping, pack and unpack, filesystem, eval, datastructures, packages, modules, objects, interfacing to the operating system, Creating Internet ware applications, Dirty Hands Internet Programming, security Issues.
UNIT - V

TCL
TCL Structure, syntax, Variables and Data in TCL, Control Flow, Data Structures, input/output, procedures, strings, patterns, files, Advance TCL- eval, source, exec and uplevel commands, Name spaces, trapping errors, event driven programs, making applications internet aware, Nuts and Bolts Internet Programming, Security Issues, C Interface.

Tk
Tk-Visual Tool Kits, Fundamental Concepts of Tk, Tk by example, Events and Binding, Perl-Tk.

Textbooks:
1. The World of Scripting Languages, David Barron, Wiley Publications.
2. Ruby Programming language by David Flanagan and Yukihiro Matsumoto O’Reilly

References:
1. Open Source Web Development with LAMP using Linux Apache, MySQL, Perl and PHP, J.Lee and B.Ware(Addison Wesley) Pearson Education.
2. Perl by Example, E. Quigley, Pearson Education.
3. Programming Perl, Larry Wall, T. Christiansen and J. Orwant, O’Reilly, SPD.
4. Tcl and the Tk Tool kit, Ousterhout, Pearson Education.
MOBILE APPLICATION DEVELOPMENT
(Professional Elective - III)

III Year B.Tech. CSE II-Semester

<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>

Prerequisites
1. Acquaintance with JAVA programming
2. A Course on “Database Management Systems”

Objectives
1. To demonstrate their understanding of the fundamentals of Android operating systems
2. To improve their skills of using Android software development tools
3. To demonstrate their ability to develop software with reasonable complexity on mobile platform
4. To demonstrate their ability to deploy software to mobile devices
5. To demonstrate their ability to debug programs running on mobile devices

Outcomes
1. Student understands the working of Android OS Practically.
2. Student will be able to develop Android user interfaces
3. Student will be able to develop, deploy and maintain the Android Applications.

UNIT - I
Introduction to Android Operating System: Android OS design and Features – Android development framework, SDK features, Installing and running applications on Android Studio, Creating AVDs, Types of Android applications, Best practices in Android programming, Android tools

Android application components – Android Manifest file, Externalizing resources like values, themes, layouts, Menus etc, Resources for different devices and languages, Runtime Configuration Changes

Android Application Lifecycle – Activities, Activity lifecycle, activity states, monitoring state changes

UNIT - II
Android User Interface: Measurements – Device and pixel density independent measuring UNIT - s Layouts – Linear, Relative, Grid and Table Layouts

User Interface (UI) Components – Editable and non editable TextViews, Buttons, Radio and Toggle Buttons, Checkboxes, Spinners, Dialog and pickers

Event Handling – Handling clicks or changes of various UI components

Fragments – Creating fragments, Lifecycle of fragments, Fragment states, Adding fragments to Activity, adding, removing and replacing fragments with fragment transactions, interfacing between fragments and Activities, Multi-screen Activities

UNIT - III
Intents and Broadcasts: Intent – Using intents to launch Activities, Explicitly starting new Activity, Implicit Intents, Passing data to Intents, Getting results from Activities, Native Actions, using Intent to dial a number or to send SMS
Broadcast Receivers – Using Intent filters to service implicit Intents, Resolving Intent filters, finding and using Intents received within an Activity

Notifications – Creating and Displaying notifications, Displaying Toasts

UNIT - IV
Persistent Storage: Files – Using application specific folders and files, creating files, reading data from files, listing contents of a directory Shared Preferences – Creating shared preferences, saving and retrieving data using Shared Preference

UNIT - V
Database – Introduction to SQLite database, creating and opening a database, creating tables, inserting retrieving and etindelg data, Registering Content Providers, Using content Providers (insert, delete, retrieve and update)

Textbooks:
1. Professional Android 4 Application Development, Reto Meier, Wiley India, (Wrox) , 2012

Reference:
1. Beginning Android 4 Application Development, Wei-Meng Lee, Wiley India (Wrox), 2013
SOFTWARE TESTING METHODOLOGIES
(Professional Elective - III)

III Year B.Tech. CSE II-Semester

Prerequisites
1. A course on “Software Engineering”

Objectives
1. To provide knowledge of the concepts in software testing such as testing process, criteria, strategies, and methodologies.
2. To develop skills in software test automation and management using latest tools.

Outcomes
1. Design and develop the best test strategies in accordance to the development model.

UNIT - I
Introduction: Purpose of testing, Dichotomies, model for testing, consequences of bugs, taxonomy of bugs
Flow graphs and Path testing: Basics concepts of path testing, predicates, path predicates and achievable paths, path sensitizing, path instrumentation, application of path testing.

UNIT - II
Transaction Flow Testing: transaction flows, transaction flow testing techniques. Dataflow testing: Basics of dataflow testing, strategies in dataflow testing, application of dataflow testing. Domain Testing: domains and paths, Nice & ugly domains, domain testing, domains and interfaces testing, domain and interface testing, domains and testability.

UNIT - III
Paths, Path products and Regular expressions: path products & path expression, reduction procedure, applications, regular expressions & flow anomaly detection.
Logic Based Testing: overview, decision tables, path expressions, kv charts, specifications.

UNIT - IV
State, State Graphs and Transition testing: state graphs, good & bad state graphs, state testing, Testability tips.

UNIT - V
Graph Matrices and Application: Motivational overview, matrix of graph, relations, power of a matrix, node reduction algorithm, building tools. (Student should be given an exposure to a tool like JMeter or Winrunner).

Textbooks

References
1. The craft of software testing, Brian Marick, Pearson Education.
2. Software Testing Techniques – SPD(Oreille)
MACHINE LEARNING LAB USING PYTHON

III Year B.Tech. CSE II-Semester

<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1.5</td>
</tr>
</tbody>
</table>

**Objective**

1. To get an overview of the various machine learning techniques and can able to demonstrate them using python.

**Outcomes**

1. Understand complexity of Machine Learning algorithms and their limitations;
2. Understand modern notions in data analysis oriented computing;
3. Be capable of confidently applying common Machine Learning algorithms in practice and implementing their own;
4. Be capable of performing experiments in Machine Learning using real-world data.

**List of Experiments**

1. The probability that it is Friday and that a student is absent is 3 %. Since there are 5 school days in a week, the probability that it is Friday is 20 %. What is the probability that a student is absent given that today is Friday? Apply Baye’s rule in python to get the result. (Ans: 15%)

2. Extract the data from database using python

3. Implement k-nearest neighbours classification using python

4. Given the following data, which specify classifications for nine combinations of VAR1 and VAR2 predict a classification for a case where VAR1=0.906 and VAR2=0.606, using the result of k-means clustering with 3 means (i.e., 3 centroids)

<table>
<thead>
<tr>
<th>VAR1</th>
<th>VAR2</th>
<th>CLASS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.713</td>
<td>1.586</td>
<td>0</td>
</tr>
<tr>
<td>0.180</td>
<td>1.786</td>
<td>1</td>
</tr>
<tr>
<td>0.353</td>
<td>1.240</td>
<td>1</td>
</tr>
<tr>
<td>0.940</td>
<td>1.566</td>
<td>0</td>
</tr>
<tr>
<td>1.486</td>
<td>0.759</td>
<td>1</td>
</tr>
<tr>
<td>1.266</td>
<td>1.106</td>
<td>0</td>
</tr>
<tr>
<td>1.540</td>
<td>0.419</td>
<td>1</td>
</tr>
<tr>
<td>0.459</td>
<td>1.799</td>
<td>1</td>
</tr>
<tr>
<td>0.773</td>
<td>0.186</td>
<td>1</td>
</tr>
</tbody>
</table>

5. The following training examples map descriptions of individuals onto high, medium and low credit-worthiness.

- medium skiing design single twenties no -> highRisk
- high golf trading married forties yes -> lowRisk
- low speedway transport married thirties yes -> medRisk
- medium football banking single thirties yes -> lowRisk
- high flying media married fifties yes -> highRisk
- low football security single twenties no -> medRisk
medium golf media single thirties yes -> medRisk
medium golf transport married forties yes -> lowRisk
high skiing banking single thirties yes -> highRisk
low golf unemployed married forties yes -> highRisk

Input attributes are (from left to right) income, recreation, job, status, age-group, home-owner. Find the unconditional probability of `golf' and the conditional probability of `single' given `medRisk' in the dataset?

6. Implement linear regression using python.
7. Implement Naïve Bayes theorem to classify the English text
8. Implement an algorithm to demonstrate the significance of genetic algorithm
9. Implement the finite words classification system using Back-propagation algorithm

Textbooks:

Reference:
COMPILER DESIGN & WEB TECHNOLOGIES LAB

III Year B.Tech. CSE II-Semester

<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>4</td>
<td>2</td>
</tr>
</tbody>
</table>

Prerequisites
1. A Course on “Objected Oriented Programming through Java”
2. A Course on “Formal Languages & Automata Theory”

Co-requisites
1. A course on “Web Technologies”

Objectives
1. To provide hands-on experience on web technologies
2. To develop client-server application using web technologies
3. To introduce server side programming with Java servlets and JSP
4. To understand the various phases in the design of a compiler.
5. To understand the design of top-down and bottom-up parsers.
6. To understand syntax directed translation schemes.
7. To introduce lex and yacc tools.

Outcomes
1. Design and develop interactive and dynamic web applications using HTML, CSS, JavaScript and XML
2. Apply client-server principles to develop scalable and enterprise web applications.
3. Ability to design, develop, and implement a compiler for any language.
4. Able to use lex and yacc tools for developing a scanner and a parser.
5. Able to design and implement LL and LR parsers.

List of Experiments

Compiler Design Experiments
1. Write a LEX Program to scan reserved word & Identifiers of C Language
2. Implement Predictive Parsing algorithm
3. Write a C program to generate three address code.
4. Implement SLR(1) Parsing algorithm
5. Design LALR bottom up parser for the given language

```plaintext
<program> ::= <block>
<block> ::= { <variabledefinition> <slist> } |
          { <slist> }
<variabledefinition> ::= int <vardeflist> ;
<vardeflist> ::= <vardec> | <vardec> , <vardeflist>
<vardec> ::= <identifier> | <identifier> [ <constant> ]
<slist> ::= <statement> | <statement> ; <slist>
<statement> ::= <assignment> | <ifstatement> | <whilestatement>
              | <block> | <printstatement> | <empty>
<assignment> ::= <identifier> = <expression>
```
| <identifier> [ <expression> ] = <expression> |
| <ifstatement> ::= if <bexpression> then <slist> else <slist> endif |
| if <bexpression> then <slist> endif |
| <whilestatement> ::= while <bexpression> do <slist> enddo |
| <printstatement> ::= print ( <expression> ) |
| <expression> ::= <expression> <addingop> <term> | <term> | <addingop> <term> |
| <bexpression> ::= <expression> <relop> <expression> |
| <relop> ::= < | <= | == | >= | > | != |
| <addingop> ::= + | - |
| <term> ::= <term> <multop> <factor> | <factor> |
| <factor> ::= <constant> | <identifier> | <identifier> [ <expression> ] |
| ( <expression> ) |
| <constant> ::= <digit> | <digit> <constant> |
| <identifier> ::= <identifier> <letterordigit> | <letter> |
| <letterordigit> ::= <letter> | <digit> |
| <letter> ::= a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z |
| <digit> ::= 0|1|2|3|4|5|6|7|8|9 |
| <empty> has the obvious meaning |

Comments (zero or more characters enclosed between the standard C/Java-style comment brackets /*...*/) can be inserted. The language has rudimentary support for 1-dimensional arrays. The declaration int a[3] declares an array of three elements, referenced as a[0], a[1] and a[2]. Note also that you should worry about the scoping of names.

A simple program written in this language is:

```
{ int a[3],t1,t2;
  t1=2;
  a[0]=1; a[1]=2; a[t1]=3;
  t2=(a[2]+t1*6)/(a[2]-t1);
  if t2>5 then
    print(t2);
  else 
    int t3;
    t3=99;
    t2=25;
    print(-t1+t2*t3); /* this is a comment on 2 lines */
} endif
```

Web Technologies Experiments:
1. Write a PHP script to print prime numbers between 1-50.
2. PHP script to
   a. Find the length of a string.
   b. Count no of words in a string.
   c. Reverse a string.
   d. Search for a specific string.
3. Write a PHP script to merge two arrays and sort them as numbers, in descending order.
4. Write a PHP script that reads data from one file and write into another file.
5. Develop static pages (using Only HTML) of an online book store. The pages should resemble: www.amazon.com. The website should consist the following pages.
   a) Home page
b) Registration and user Login  
c) User Profile Page  
d) Books catalog  
e) Shopping Cart  
f) Payment By credit card  
g) Order Confirmation

6. Validate the Registration, user login, user profile and payment by credit card pages using JavaScript.
7. Create and save an XML document on the server, which contains 10 users information. Write a program, which takes User Id as an input and returns the user details by taking the user information from the XML document.
8. Install TOMCAT web server. Convert the static web pages of assignments 2 into dynamic web pages using servlets and cookies. Hint: Users information (user id, password, credit card number) would be stored in web.xml. Each user should have a separate Shopping Cart.
9. Redo the previous task using JSP by converting the static web pages of assignments 2 into dynamic web pages. Create a database with user information and books information. The books catalogue should be dynamically loaded from the database. Follow the MVC architecture while doing the website.

Textbooks:
1. WEB TECHNOLOGIES: A Computer Science Perspective, Jeffrey C. Jackson, Pearson Education

References:
### III Year B.Tech. CSE II-Semester

<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1.5</td>
</tr>
</tbody>
</table>

1. Design and implement Two-thread mutual exclusion algorithm (Peterson’s Algorithm) using multithreaded programming.
2. Design and implement Filter Lock algorithm and check for deadlock-free and starvation-free conditions using multithreaded programming.
3. Design and implement Lamport’s Bakery Algorithm and check for deadlock-free and starvation-free conditions using multithreaded programming.
4. Design and implement Lock-based concurrent FIFO queue data structure using multithreaded programming.
5. Design a consensus object using read–write registers by implementing a deadlock-free or starvation-free mutual exclusion lock. (Use CompareAndSet() Primitive).
6. Design and implement concurrent List queue data structure using multithreaded programming. (Use Atomic Primitives)
7. Design and implement concurrent Stack queue data structure using multithreaded programming. (Use Atomic Primitives)
8. Design and implement concurrent FIFO queue data structure using multithreaded programming. (Use Atomic Primitives)

### Textbooks:


### References:

/network programming lab

(Professional Elective - III)

III Year B.Tech. CSE II-Semester

Objectives

1. To understand inter process and inter-system communication
2. To understand socket programming in its entirety
3. To understand usage of TCP/UDP / Raw sockets
4. To understand how to build network applications

Outcomes

1. To write socket API based programs
2. To design and implement client-server applications using TCP and UDP sockets
3. To analyze network programs

List of Experiments

1. Implement programs for Inter Process Communication using PIPE, Message Queue and Shared Memory.
2. Write a programme to create an integer variable using shared memory concept and increment the variable simultaneously by two processes. Use semaphores to avoid race conditions.
3. Design TCP iterative Client and server application to reverse the given input sentence
4. Design TCP iterative Client and server application to reverse the given input sentence
5. Design TCP client and server application to transfer file
6. Design a TCP concurrent server to convert a given text into upper case using multiplexing system call “select”
7. Design a TCP concurrent server to echo given set of sentences using poll functions
8. Design UDP Client and server application to reverse the given input sentence
9. Design UDP Client server to transfer a file
10. Design using poll client server application to multiplex TCP and UDP requests for converting a given text into upper case.
11. Design a RPC application to add and subtract a given pair of integers

Textbooks:

SCRIPTING LANGUAGES LAB
(Professional Elective - III)

III Year B.Tech. CSE II-Semester

Prerequisites: Any High level programming language (C,C++)

Objectives
1. To Understand the concepts of scripting languages for developing web based projects
2. To understand the applications the of Ruby, TCL, Perl scripting languages

Outcomes
1. Ability to understand the differences between Scripting languages and programming languages
2. Able to gain some fluency programming in Ruby, Perl, TCL

List of Experiments
1. Write a Ruby script to create a new string which is n copies of a given string where n is a non-negative integer
2. Write a Ruby script which accept the radius of a circle from the user and compute the parameter and area.
3. Write a Ruby script which accept the user's first and last name and print them in reverse order with a space between them
4. Write a Ruby script to accept a filename from the user print the extension of that
5. Write a Ruby script to find the greatest of three numbers
6. Write a Ruby script to print odd numbers from 10 to 1
7. Write a Ruby script to check two integers and return true if one of them is 20 otherwise return their sum
8. Write a Ruby script to check two temperatures and return true if one is less than 0 and the other is greater than 100
9. Write a Ruby script to print the elements of a given array
10. Write a Ruby program to retrieve the total marks where subject name and marks of a student stored in a hash
11. Write a TCL script to find the factorial of a number
12. Write a TCL script that multiplies the numbers from 1 to 10
13. Write a TCL script for Sorting a list using a comparison function
14. Write a TCL script to (i) create a list (ii) append elements to the list (iii) Traverse the list (iv) Concatenate the list
15. Write a TCL script to comparing the file modified times.
16. Write a TCL script to Copy a file and translate to native format.
17. a) Write a Perl script to find the largest number among three numbers.
   b) Write a Perl script to Copy a file and translate to native format.
18. Write a Perl program to implement the following list of manipulating functions
   a) Shift
   b) Unshift
   c) Push
19. a) Write a Perl script to substitute a word, with another word in a string.
    b) Write a Perl script to validate IP address and email address.
20. Write a Perl script to print the file in reverse order using command line arguments
Textbooks:
1. The World of Scripting Languages, David Barron, Wiley Publications.
2. Ruby Programming language by David Flanagan and Yukihiro Matsumoto O’Reilly

References:
1. Open Source Web Development with LAMP using Linux Apache, MySQL, Perl and PHP, J.Lee and B.Ware (Addison Wesley) Pearson Education.
2. Perl by Example, E. Quigley, Pearson Education.
3. Programming Perl, Larry Wall, T. Christiansen and J. Orwant, O’Reilly, SPD.
4. Tcl and the Tk Tool kit, Ousterhout, Pearson Education.
MOBILE APPLICATION DEVELOPMENT LAB
(Professional Elective - III)

III Year B.Tech. CSE II-Semester

<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1.5</td>
</tr>
</tbody>
</table>

Prerequisites: --- NIL---

Objectives

1. To learn how to develop Applications in android environment.
2. To learn how to develop user interface applications.
3. To learn how to develop URL related applications.

Outcomes

1. Student understands the working of Android OS Practically.
2. Student will be able to develop user interfaces.
3. Student will be able to develop, deploy and maintain the Android Applications.

List of Experiments

1. Create an Android application that shows Hello + name of the user and run it on an emulator. (b)
   Create an application that takes the name from a text box and shows hello message along with the
   name entered in text box, when the user clicks the OK button.

2. Create a screen that has input boxes for User Name, Password, Address, Gender (radio buttons for
   male and female), Age (numeric), Date of Birth (Date Picket), State (Spinner) and a Submit button.
   On clicking the submit button, print all the data below the Submit Button. Use (a) Linear Layout (b)
   Relative Layout and (c) Grid Layout or Table Layout.

3. Develop an application that shows names as a list and on selecting a name it should show the
   details of the candidate on the next screen with a “Back” button. If the screen is rotated to landscape
   mode (width greater than height), then the screen should show list on left fragment and details on
   right fragment instead of second screen with back button. Use Fragment transactions and Rotation
   event listener.

4. Develop an application that uses a menu with 3 options for dialing a number, opening a website and
   to send an SMS. On selecting an option, the appropriate action should be invoked using intents.

5. Develop an application that inserts some notifications into Notification area and whenever a
   notification is inserted, it should show a toast with details of the notification.

6. Create an application that uses a text file to store user names and passwords (tab separated fields
   and one record per line). When the user submits a login name and password through a screen, the
   details should be verified with the text file data and if they match, show a dialog saying that login is
   successful. Otherwise, show the dialog with Login Failed message.

7. Create a user registration application that stores the user details in a database table.
8. Create a database and a user table where the details of login names and passwords are stored. Insert some names and passwords initially. Now the login details entered by the user should be verified with the database and an appropriate dialog should be shown to the user.

9. Create an admin application for the user table, which shows all records as a list and the admin can select any record for edit or modify. The results should be reflected in the table.

10. Develop an application that shows all contacts of the phone along with details like name, phone number, mobile number etc.

11. Create an application that saves user information like name, age, gender etc. in shared preference and retrieves them when the program restarts.

12. Create an alarm that rings every Sunday at 8:00 AM. Modify it to use a time picker to set alarm time.

13. Create an application that shows the given URL (from a text field) in a browser.

Textbooks:
1. Professional Android 4 Application Development, Reto Meier, Wiley India, (Wrox) , 2012

Reference:
1. Beginning Android 4 Application Development, Wei-Meng Lee, Wiley India (Wrox), 2013
SOFTWARE TESTING METHODOLOGIES LAB  
(Professional Elective - III)

III Year B.Tech. CSE II-Semester 
L   T    P   C
0    0   3  1.5

Prerequisites
1. A basic knowledge of programming.

Objectives
1. To provide knowledge of Software Testing Methods.
2. To develop skills in software test automation and management using latest tools.

Outcome
1. Design and develop the best test strategies in accordance to the development model.

List of Experiments

1. Recording in context sensitive mode and analog mode
2. GUI checkpoint for single property
3. GUI checkpoint for single object/window
4. GUI checkpoint for multiple objects
5. a) Bitmap checkpoint for object/window
    b) Bitmap checkpoint for screen area
6. Database checkpoint for Default check
7. Database checkpoint for custom check
8. Database checkpoint for runtime record check
9. a) Data driven test for dynamic test data submission
    b) Data driven test through flat files
    c) Data driven test through front grids
    d) Data driven test through excel test
10. a) Batch testing without parameter passing
    b) Batch testing with parameter passing
11. Data driven batch
12. Silent mode test execution without any interruption
13. Test case for calculator in windows application

Textbooks:
References:

1. The craft of software testing, Brian Marick, Pearson Education.
2. Software Testing Techniques – SPD(Oreille)
INFORMATION SECURITY

IV Year B.Tech. CSE I-Semester

Prerequisites
1. A Course on “Computer Networks”
2. A Course on “Mathematics”

Objectives
1. To understand the fundamentals of Cryptography
2. To understand various key distribution and management schemes
3. To understand how to deploy encryption techniques to secure data in transit across data networks
4. To apply algorithms used for secure transactions in real world applications

Outcomes
1. Demonstrate the knowledge of cryptography, network security concepts and applications.
2. Ability to apply security principles in system design.
3. Ability to identify and investigate vulnerabilities and security threats and mechanisms to counter them.

UNIT - I
Security Attacks (Interruption, Interception, Modification and Fabrication), Security Services (Confidentiality, Authentication, Integrity, Non-repudiation, access Control and Availability) and Mechanisms, A model for Internetwork security.


UNIT - II
Message authentication and Hash Functions, Authentication Requirements and Functions, Message Authentication, Hash Functions and MACs Hash and MAC Algorithms SHA-512, HMAC.

UNIT - III
Email Security: Pretty Good Privacy (PGP) and S/MIME.

UNIT - IV
IP Security:

UNIT - V

Textbook:

References:
BIG DATA ANALYTICS

IV Year B.Tech. CSE I-Semester

<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
</tbody>
</table>

Prerequisites
1. A Course on “Data Mining”

Objectives
1. The purpose of this course is to provide the students with the knowledge of Big data Analytics principles and techniques.
2. This course is also designed to give an exposure of the frontiers of Big data Analytics

Outcomes
1. Ability to explain the foundations, definitions, and challenges of Big Data and various Analytical tools.
2. Ability to program using HADOOP and Map reduce, NOSQL
3. Ability to understand importance of Big Data in Social Media and Mining.

UNIT - I

UNIT - II
Big Data Technologies : Hadoop’s Parallel World – Data discovery – Open source technology for Big Data Analytics – cloud and Big Data – Predictive Analytics – Mobile Business Intelligence and Big Data

UNIT - III
Introduction Hadoop : Big Data – Apache Hadoop & HadoopEcoSystem – Moving Data in and out of Hadoop – Understanding inputs and outputs of MapReduce - Data Serialization.

UNIT - IV
Hadoop Architecture : Hadoop RDBMS Vs Hadoop, Hadoop Overview, Hadoop distributors, HDFS, HDFS Daemons, Anatomy of File Write and Read, NameNode, Secondary NameNode, and DataNode, HDFS Architecture, Hadoop Configuration, Map Reduce Framework, Role of HBase in Big Data processing, HIVE, PIG.

UNIT - V
Data Analytics with R Machine Learning : Introduction, Supervised Learning, Unsupervised Learning, Collaborative Filtering, Social Media Analytics, Mobile Analytics, Big Data Analytics with BigR.

Textbooks:
References:

GRAPH THEORY
(Professional Elective - IV)

IV Year B.Tech. CSE I-Semester

Prerequisites

1. An understanding of Mathematics in general is sufficient.

Outcomes:

1. Know some important classes of graph theoretic problems;
2. Be able to formulate and prove central theorems about trees, matching, connectivity, colouring and planar graphs;
3. Be able to describe and apply some basic algorithms for graphs;
4. Be able to use graph theory as a modelling tool.

UNIT - I
Introduction-Discovery of graphs, Definitions, Subgraphs, Isomorphic graphs, Matrix representations of graphs, Degree of a vertex, Directed walks, paths and cycles, Connectivity in digraphs, Eulerian and Hamilton digraphs, Eulerian digraphs, Hamilton digraphs, Special graphs, Complements, Larger graphs from smaller graphs, Union, Sum, Cartesian Product, Composition, Graphic sequences, Graph theoretic model of the LAN problem, Havel-Hakimi criterion, Realization of a graphic sequence.

UNIT - II
Connected graphs and shortest paths - Walks, trails, paths, cycles, Connected graphs, Distance, Cut-vertices and cut-edges, Blocks, Connectivity, Weighted graphs and shortest paths, Weighted graphs, Dijkstra’s shortest path algorithm, Floyd-Warshall shortest path algorithm.

UNIT - III
Trees- Definitions and characterizations, Number of trees, Cayley’s formula, Kircho-matrix-tree theorem, Minimum spanning trees, Kruskal’s algorithm, Prim’s algorithm, Special classes of graphs, Bipartite Graphs, Line Graphs, Chordal Graphs, Eulerian Graphs, Fleury’s algorithm, Chinese Postman problem, Hamilton Graphs, Introduction, Necessary conditions and sufficient conditions.

UNIT - IV
Independent sets coverings and matchings- Introduction, Independent sets and coverings: basic equations, Matchings in bipartite graphs, Hall’s Theorem, Konig’s Theorem, Perfect matchings in graphs, Greedy and approximation algorithms.

UNIT - V

Textbooks:

2. Graph Theory with Applications, J. A. Bondy and U. S. R. Murty, Elsevier.
References:

1. Lecture Videos: http://nptel.ac.in/courses/111106050/13
2. Introduction To Graph Theory, Douglas B. West, Pearson.
3. Schaum's Outlines Graph Theory, Balakrishnan ,TMH
4. Introduction to Graph Theory, Wilson Robin j, PHI
5. Graph Theory With Applications To Engineering And Computer Science ,Narsing Deo,PHI
EMBEDDED SYSTEMS
(Professional Elective - IV)

IV Year B.Tech. CSE I-Semester

<table>
<thead>
<tr>
<th></th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>

Prerequisites
1. A course on “Digital Logic Design and Microprocessors”
2. A course on “Computer Organization and Architecture”

Objectives
1. To provide an overview of principles of Embedded System
2. To provide a clear understanding of role of firmware, operating systems in correlation with hardware systems.

Outcomes
1. Expected to understand the selection procedure of processors in the embedded domain.
2. Design procedure of embedded firm ware.
3. Expected to visualize the role of realtime operating systems in embedded systems.
4. Expected to evaluate the correlation between task synchronization and latency issues

UNIT - I
Introduction to Embedded Systems:

UNIT - II
The Typical Embedded System:
Core of the Embedded System, Memory, Sensors and Actuators, Communication Interface, Embedded Firmware, Other System components.

UNIT - III
Embedded Firmware Design and Development:
Embedded Firmware Design, Embedded Firmware Development Languages, Programming in Embedded C.

UNIT - IV
RTOS Based Embedded System Design:
Operating System basics, Types of Operating Systems, Tasks, Process, Threads, Multiprocessing and Multi tasking, Task Scheduling, Threads-Processes-Scheduling putting them together, Task Communication, Task Synchronization, Device Drivers, How to choose an RTOS.

UNIT - V
Integration and Testing of Embedded Hardware and Firmware:
Integration of Hardware and Firmware, Boards Bring up
The Embedded System Development Environment:
The Integrated Development Environment(IDE), Types of files generated on Cross-Compilation, Disassembler/Decompiler, Simulators, Emulators and Debugging, Target Hardware Debugging, Boundary Scan.
Textbooks:

References:
2. Embedded Systems Design - A Unified Hardware/Software Introduction, Frank Vahid and Tony Givargis, John Wiley
3. Embedded Systems, Lyla, Pearson
SEMANTIC WEB
(Professional Elective – IV)

IV Year B.Tech. CSE I-Semester

<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>

Objectives
1. To learn Web Intelligence
2. To learn Knowledge Representation for the Semantic Web
3. To learn Ontology Engineering
4. To learn Semantic Web Applications, Services and Technology

Outcomes
1. Ability to understand Semantic Web
2. Ability to learn SOAP, UDDI
3. Ability to handle multiple web services using Orchestration
4. Ability to experiment with XML Technologies
5. Ability to construct and use Ontologies

UNIT - I

UNIT - II

UNIT - III
Resource Description Framework: Features, Capturing Knowledge with RDF.

XML Technologies: XPath, The Style Sheet Family: XSL, XSLT, and XSLFO, XQuery, XLink, XPointer, XInclude, XMLBase, XHTML, XForms, SVG.

UNIT - IV

UNIT - V

Textbooks:

2. Thinking on the Web - Berners Lee, Godel and Turing, Wiley Interscience

References:

4. Programming the Semantic Web, T.Segaran, C.Evans, J.Taylor, O'Reilly, SPD.
CLOUD COMPUTING
(Professional Elective - IV)

IV Year B.Tech. CSE I-Semester

<table>
<thead>
<tr>
<th></th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syllabus</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>

Prerequisites
1. A course on “Computer Networks”
2. A course on “Operating Systems”
3. A course on “Distributed Systems”

Objectives
1. This course provides an insight into cloud computing
2. Topics covered include- distributed system models, different cloud service models, service oriented architectures, cloud programming and software environments, resource management.

Outcomes
1. Ability to understand various service delivery models of a cloud computing architecture.
2. Ability to understand the ways in which the cloud can be programmed and deployed.
3. Understanding cloud service providers.

UNIT - I

UNIT - II

UNIT - III

UNIT - IV

UNIT - V
Cloud Service Providers: EMC, EMC IT, Captiva Cloud Toolkit, Google, Cloud Platform, Cloud Storage, Google Cloud Connect, Google Cloud Print, Google App Engine, Amazon Web Services, Amazon Elastic Compute Cloud, Amazon Simple Storage Service, Amazon Simple Queue, service, Microsoft, Windows Azure, Microsoft Assessment and Planning Toolkit, SharePoint, IBM, Cloud Models, IBM Smart Cloud,
SAP Labs, SAP HANA Cloud Platform, Virtualization Services Provided by SAP, Sales force, Sales Cloud, Service Cloud: Knowledge as a Service, Rack space, VMware, Manjra soft, Aneka Platform

Textbook:


References:

DISTRIBUTED SYSTEMS
(Professional Elective – IV)

IV Year B.Tech. CSE I-Semester

<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>

Prerequisites
1. A course on “ Operating Systems”
2. A course on “Computer Organization & Architecture”

Objectives
1. This course provides an insight into Distributed systems.
2. Topics include- Peer to Peer Systems, Transactions and Concurrency control, Security and Distributed shared memory

Outcomes
1. Ability to understand Transactions and Concurrency control.
2. Ability to understand Security issues.
3. Understanding Distributed shared memory.
4. Ability to design distributed systems for basic level applications.

UNIT - I
Characterization of Distributed Systems-Introduction, Examples of Distributed systems, Resource sharing and web, challenges, System models-Introduction ,Architectural and Fundamental models, Networking and Internetworking, Interprocess Communication, Distributed objects and Remote Invocation-Introduction, Communication between distributed objects, RPC, Events and notifications, Case study-Java RMI.

UNIT - II

UNIT - III

UNIT - IV
Transactions and Concurrency control-Introduction, Transactions, Nested Transactions, Locks, Optimistic concurrency control, Timestamp ordering. Distributed Transactions-Introduction, Flat and Nested Distributed Transactions, Atomic commit protocols, Concurrency control in distributed transactions, Distributed deadlocks, Transaction recovery.

UNIT - V
Replication-Introduction, System model and group communication, Fault tolerant services, Transactions with replicated data. Distributed shared memory, Design and Implementation issues, Consistency models.
Textbooks:


References:

ADVANCED ALGORITHMS
(Professional Elective - V)

IV Year B.Tech. CSE I-Semester

<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>

Prerequisites
1. A course on “Computer Programming & Data Structures”
2. A course on “Advanced Data Structures & Algorithms”

Objectives
1. Introduces the recurrence relations for analyzing the algorithms
2. Introduces the graphs and their traversals.
3. Describes major algorithmic techniques (divide-and-conquer, greedy, dynamic programming, Brute Force, Transform and Conquer approaches) and mention problems for which each technique is appropriate;
4. Describes how to evaluate and compare different algorithms using worst-case, average-case and best-case analysis.
5. Introduces string matching algorithms
6. Introduces linear programming.

Outcomes
1. Ability to analyze the performance of algorithms
2. Ability to choose appropriate data structures and algorithm design methods for a specified application
3. Ability to understand how the choice of data structures and the algorithm design methods impact the performance of programs

UNIT - I


UNIT - II
Greedy Algorithms - Huffman Codes, Activity Selection Problem. Amortized Analysis.

Graph Algorithms: Topological Sorting, Minimum Spanning trees, Single Source Shortest Paths, Maximum Flow algorithms..

UNIT - III
Sorting Networks: Comparison Networks, Zero-one principle, bitonic Sorting Networks, Merging Network, Sorting Network.

Matrix Operations- Strassen's Matrix Multiplication, Inverting matrices, Solving system of linear Equations
UNIT - IV

UNIT- V
NP-Completeness and Approximation Algorithms: Polynomial time, polynomial time verification, NP-Completeness and reducibility, NP-Complete problems. Approximation Algorithms- Vertex cover Problem, Travelling Sales person problem

Textbook:


References:

2. Design and Analysis Algorithms - Parag Himanshu Dave, Himanshu Bhalchandra Dave, Pearson
MOBILE COMPUTING
(Professional Elective - V)

IV Year B.Tech. CSE I-Semester

Prerequisites
1. A course on “Computer Networks”

Objectives
1. To make the student understand the concept of mobile computing paradigm, its novel applications and limitations.
2. To understand the typical mobile networking infrastructure through a popular GSM protocol
3. To understand the issues and solutions of various layers of mobile networks, namely MAC layer, Network Layer & Transport Layer
4. To understand the database issues in mobile environments & data delivery models.
5. To understand the ad hoc networks and related concepts.
6. To understand the platforms and protocols used in the mobile environment.

Outcomes
1. Able to think and develop new mobile application.
2. Able to take any new technical issue related to this new paradigm and come up with a solution(s).
3. Able to develop new ad hoc network applications and/or algorithms/protocols.
4. Able to understand & develop any existing or new protocol related to the mobile environment.

UNIT - I
Introduction
Mobile Communications, Mobile Computing – Paradigm, Promises/Novel Applications and Impediments and Architecture; Mobile and Handheld Devices, Limitations of Mobile and Handheld Devices.
GSM – Services, System Architecture, Radio Interfaces, Protocols, Localization, Calling, Handover, Security, New Data Services, GPRS.

UNIT - II
(Wireless) Medium Access Control (MAC)
Motivation for a specialized MAC (Hidden and exposed terminals), SDMA, FDMA, TDMA, CDMA, Wireless LAN/(IEEE 802.11)

Mobile Network Layer
IP and Mobile IP Network Layers, Packet Delivery and Handover Management, Location Management, Registration, Tunneling and Encapsulation, Route Optimization, DHCP.

UNIT - III
Mobile Transport Layer

Database Issues
Database Hoarding & Caching Techniques, Client-Server Computing & Adaptation, Transactional Models.
UNIT - IV
Data Dissemination and Synchronization
Communications Asymmetry, Classification of Data Delivery Mechanisms, Data Dissemination, Broadcast Models, Selective Tuning and Indexing Methods.

UNIT - V
Mobile Ad hoc Networks (MANETs)
Introduction, Applications & Challenges of a MANET, Routing, Classification of Routing Algorithms, Algorithms such as DSR, AODV, DSDV, Mobile Agents, Service Discovery.

Textbooks:
SOFT COMPUTING  
(Professional Elective - V)  

IV Year B.Tech. CSE I-Semester  

<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>

Objectives  
1. Familiarize with soft computing concepts  
2. Introduce and use the idea of fuzzy logic and use of heuristics based on human experience  
3. Familiarize the Neuro-Fuzzy modeling using Classification and Clustering techniques  
4. Learn the concepts of Genetic algorithm and its applications  
5. Acquire the knowledge of Rough Sets.

Outcomes  
1. Identify the difference between Conventional Artificial Intelligence to Computational Intelligence.  
2. Understand fuzzy logic and reasoning to handle and solve engineering problems  
3. Apply the Classification and clustering techniques on various applications.  
4. Understand the advanced neural networks and its applications  
5. Perform various operations of genetic algorithms, Rough Sets.  
6. Comprehend various techniques to build model for various applications

UNIT - I  

UNIT - II  

UNIT - III  
Fuzzy Decision Making, Particle Swarm Optimization.

UNIT - IV  

UNIT - V  
Rough Sets, Rough Sets, Rule Induction, and Discernibility Matrix, Integration of Soft Computing Techniques.

Textbook:  
References:

2. Genetic Algorithms-In Search, optimization and Machine learning, David E. Goldberg, Pearson Education.
INTERNET OF THINGS
(Professional Elective - V)

IV Year B.Tech. CSE I-Semester

Objectives
1. To introduce the terminology, technology and its applications
2. To introduce the concept of M2M (machine to machine) with necessary protocols
3. To introduce the Python Scripting Language which is used in many IoT devices
4. To introduce the Raspberry PI platform, that is widely used in IoT applications
5. To introduce the implementation of web based services on IoT devices

Outcomes
1. Interpret the impact and challenges posed by IoT networks leading to new architectural models.
2. Compare and contrast the deployment of smart objects and the technologies to connect them to network.
3. Appraise the role of IoT protocols for efficient network communication.
4. Elaborate the need for Data Analytics and Security in IoT.
5. Illustrate different sensor technologies for sensing real world entities and identify the applications of IoT in Industry.

UNIT - I

UNIT - II
IoT and M2M – Software defined networks, network function virtualization, difference between SDN and NFV for IoT Basics of IoT System Management with NETCOZF, YANG- NETCONF, YANG, SNMP NETOPEER

UNIT - III
Introduction to Python - Language features of Python, Data types, data structures, Control of flow, functions, modules, packaging, file handling, data/time operations, classes, Exception handling Python packages - JSON, XML, HTTPLib, URLLib, SMTPLib

UNIT - IV
IoT Physical Devices and Endpoints - Introduction to Raspberry PI-Interfaces (serial, SPI, I2C) Programming – Python program with Raspberry PI with focus of interfacing external gadgets, controlling output, reading input from pins.

UNIT - V
IoT Physical Servers and Cloud Offerings – Introduction to Cloud Storage models and communication APIs Webserver – Web server for IoT, Cloud for IoT, Python web application framework Designing a RESTful web API
Textbooks:


SOFTWARE PROCESS & PROJECT MANAGEMENT
(Professional Elective - V)

IV Year B.Tech. CSE I-Semester

<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>

**Objectives**

1. To acquire knowledge on software process management
2. To acquire managerial skills for software project development
3. To understand software economics

**Outcomes**

1. Gain knowledge of software economics, phases in the life cycle of software development, project organization, project control and process instrumentation
2. Analyze the major and minor milestones, artifacts and metrics from management and technical perspective
3. Design and develop software product using conventional and modern principles of software project management

**UNIT - I**
Software Process Maturity
Process Reference Models
Capability Maturity Model (CMM), CMMI, PCMM, PSP, TSP).

**UNIT - II**
Software Project Management Renaissance
Life-Cycle Phases and Process artifacts
Engineering and Production stages, inception phase, elaboration phase, construction phase, transition phase, artifact sets, management artifacts, engineering artifacts and pragmatic artifacts, model based software architectures.

**UNIT - III**
Workflows and Checkpoints of process
Software process workflows, Iteration workflows, Major milestones, minor milestones, periodic status assessments.
Process Planning
Work breakdown structures, Planning guidelines, cost and schedule estimating process, iteration planning process, Pragmatic planning.

**UNIT - IV**
Project Organizations
Line-of- business organizations, project organizations, evolution of organizations, process automation.
Project Control and process instrumentation
The seven core metrics, management indicators, quality indicators, life-cycle expectations, Pragmatic software metrics, metrics automation.
UNIT - V
CCPDS-R Case Study and Future Software Project Management Practices

Textbooks:
1. Managing the Software Process, Watts S. Humphrey, Pearson Education
2. Software Project Management, Walker Royce, Pearson Education

References:
5. Head First PMP, Jennifer Greene & Andrew Stellman, O’Reilly, 2007
7. Agile Project Management, Jim Highsmith, Pearson education, 2004
BIG DATA ANALYTICS LAB

IV Year B.Tech. CSE I-Semester

L T P C
0 0 2 1

Objectives
1. The purpose of this course is to provide the students with the knowledge of Big data Analytics principles and techniques.
2. This course is also designed to give an exposure of the frontiers of Big data Analytics

Outcomes
1. Ability to explain the foundations, definitions, and challenges of Big Data and various Analytical tools.
2. Ability to program using HADOOP and Map reduce, NOSQL
3. Ability to understand importance of Big Data in Social Media and Mining.

List of Experiments
1. Implement a simple map-reduce job that builds an inverted index on the set of input documents (Hadoop)
2. Perform Social media analysis using cassandra
3. Buyer event analytics using Cassandra on suitable product sales data.
4. Using Power Pivot (Excel) Perform the following on any dataset
   a) Big Data Analysis
   b) Big Data Charting
5. Use R-Project to carry out statistical analysis of big data
6. Use R-Project for data visualization of social media data

Textbooks:

References:
MANAGEMENT FUNDAMENTALS FOR ENGINEERS

IV Year B.Tech. CSE II-Semester

Objective

1. To understand the Management Concepts, applications of Concepts in Practical aspects of business and development of Managerial Skills for Engineers.

Outcomes

1. The students understand the significance of Management in their Profession. The various Management Functions like Planning, Organizing, Staffing, Leading, Motivation and Control aspects are learnt in this course.
2. The students can explore the Management Practices in their domain area.

UNIT - I
Introduction to Management: Evolution of Management, Nature & Scope-Functions of Management-Role of Manager-levels of Management-Managerial Skills - Challenges-Planning-Planning Process-Types of Plans-MBO

UNIT - II
Organization Structure & HRM: Organization Design-Organizational Structure-Departmentation-Delegation-Centralization- Decentralization-Recentralization-Organizational Culture- Organizational climate- Organizational change

UNIT - III

UNIT - IV

UNIT - V
Textbooks:

COMPUTATIONAL COMPLEXITY
(Professional Elective - VI)

IV Year B.Tech. CSE II-Semester

Prerequisites
1. A course on “Computer Programming and Data Structures”
2. A course on “Discrete Structures and Graph Theory”

Objectives
1. Introduces to theory of computational complexity classes
2. Discuss about algorithmic techniques and application of these techniques to problems.
3. Introduce to randomized algorithms and discuss how effective they are in reducing time and space complexity.
4. Discuss about Graph based algorithms and approximation algorithms
5. Discuss about search trees

Outcomes
1. Ability to classify decision problems into appropriate complexity classes
2. Ability to specify what it means to reduce one problem to another, and construct reductions for simple examples.
3. Ability to classify optimization problems into appropriate approximation complexity classes
4. Ability to choose appropriate data structure for the given problem
5. Ability to choose and apply appropriate design method for the given problem

UNIT - I
Computational Complexity: Polynomial time and its justification, Nontrivial examples of polynomial-time algorithms, the concept of reduction (reducibility), Class P Class NP and NP- Completeness, The P versus NP problem and why it’s hard

UNIT - II

UNIT - III
Randomized Algorithms: Finger Printing, Pattern Matching, Graph Problems, Algebraic Methods, Probabilistic Primality Testing, De-Randomization Advanced Algorithms;

UNIT - IV

UNIT - V
Advanced Data Structures and applications: Decision Trees and Circuits, B-Trees, AVL Trees, Red and Black trees, Dictionaries and tries, Maps, Binomial Heaps, Fibonacci Heaps, Disjoint sets, Union by Rank and Path Compression
Textbooks:

AD HOC & SENSOR NETWORKS
(Professional Elective - VI)

IV Year B.Tech. CSE II-Semester

<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>

Prerequisites
1. A course on “Computer Networks”
2. A course on “Mobile Computing”

Objectives
1. To understand the concepts of sensor networks
2. To understand the MAC and transport protocols for ad hoc networks
3. To understand the security of sensor networks
4. To understand the applications of ad hoc and sensor networks

Outcomes
1. Ability to understand the state of the art research in the emerging subject of Ad Hoc and Wireless Sensor Networks
2. Ability to solve the issues in real-time application development based on ASN.
3. Ability to conduct further research in the domain of ASN

UNIT - I
Introduction to Ad Hoc Networks - Characteristics of MANETs, Applications of MANETs and Challenges of MANETs.
Routing in MANETs - Criteria for classification, Taxonomy of MANET routing algorithms, Topology-based routing algorithms-Proactive: DSDV; Reactive: DSR, AODV; Hybrid: ZRP; Position-based routing algorithms-Location Services-DREAM, Quorum-based; Forwarding Strategies: Greedy Packet, Restricted Directional Flooding-DREAM, LAR.

UNIT - II
Data Transmission - Broadcast Storm Problem, Rebroadcasting Schemes-Simple-flooding, Probability-based Methods, Area-based Methods, Neighbor Knowledge-based: SBA, Multipoint Relaying, AHBP. Multicasting: Tree-based: AMRIS, MAODV; Mesh-based: ODMRP, CAMP; Hybrid: AMRoute, MCEDAR.

UNIT - III
Geocasting: Data-transmission Oriented-LBM; Route Creation Oriented-GeoTORA, MGR. TCP over Ad Hoc TCP protocol overview, TCP and MANETs, Solutions for TCP over Ad hoc

UNIT - IV
Basics of Wireless, Sensors and Lower Layer Issues
Applications, Classification of sensor networks, Architecture of sensor network, Physical layer, MAC layer, Link layer, Routing Layer.

UNIT - V
Upper Layer Issues of WSN
Transport layer, High-level application layer support, Adapting to the inherent dynamic nature of WSNs, Sensor Networks and mobile robots.
Textbooks:

NEURAL NETWORKS & DEEP LEARNING
(Professional Elective - VI)

IV Year B.Tech. CSE II-Semester

<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>

Objectives
1. To introduce the foundations of Artificial Neural Networks
2. To acquire the knowledge on Deep Learning Concepts
3. To learn various types of Artificial Neural Networks
4. To gain knowledge to apply optimization strategies

Outcomes
1. Ability to understand the concepts of Neural Networks
2. Ability to select the Learning Networks in modeling real world systems
3. Ability to use an efficient algorithm for Deep Models
4. Ability to apply optimization strategies for large scale applications

UNIT - I
Artificial Neural Networks

UNIT - II

UNIT - III
Introduction to Deep Learning, Historical Trends in Deep learning, Deep Feed - forward networks, Gradient-Based learning, Hidden Units, Architecture Design, Back-Propagation and Other Differentiation Algorithms

UNIT - IV
Regularization for Deep Learning
Parameter norm Penalties, Norm Penalties as Constrained Optimization, Regularization and Under-Constrained Problems, Dataset Augmentation, Noise Robustness, Semi-Supervised learning, Multi-task learning, Early Stopping, Parameter Typing and Parameter Sharing, Sparse Representations, Bagging and other Ensemble Methods, Dropout, Adversarial Training, Tangent Distance, tangent Prop and Manifold, Tangent Classifier

UNIT - V
Optimization for Train Deep Models

Applications
Large-Scale Deep Learning, Computer Vision, Speech Recognition, Natural Language Processing

Textbooks:
1. Deep Learning: An MIT Press Book By Ian Goodfellow and Yoshua Bengio and Aaron Courville
CYBER FORENSICS
(Professional Elective - VI)

IV Year B.Tech. CSE II-Semester

L    T    P    C
3    0    0    3

Prerequisites
1. A Course on “Network Security”

Objectives
1. A brief explanation of the objective is to provide digital evidences which are obtained from digital media.
2. In order to understand the objectives of computer forensics, first of all, people have to recognize the different roles computer plays in a certain crime.
3. According to a snippet from the United States Security Service, the functions computer has in different kinds of crimes.

Outcomes
1. Students will understand the usage of computers in forensic, and how to use various forensic tools for a wide variety of investigations.
2. It gives an opportunity to students to continue their zeal in research in computer forensics

UNIT - I

UNIT - II
Initial Response and forensic duplication, Initial Response & Volatile Data Collection from Windows system - Initial Response & Volatile Data Collection from Unix system – Forensic Duplication: Forensic duplicates as Admissible Evidence, Forensic Duplication Tool Requirements, Creating a Forensic. Duplicate/Qualified Forensic Duplicate of a Hard Drive

UNIT - III
Forensics analysis and validation: Determining what data to collect and analyze, validating forensic data, addressing data-hiding techniques, performing remote acquisitions

Network Forensics: Network forensics overview, performing live acquisitions, developing standard procedures for network forensics, using network tools, examining the honeynet project.

UNIT - IV
Current Forensic tools: evaluating computer forensic tool needs, computer forensics software tools, computer forensics hardware tools, validating and testing forensics software E-Mail Investigations: Exploring the role of e-mail in investigation, exploring the roles of the client and server in e-mail, investigating e-mail crimes and violations, understanding e-mail servers, using specialized e-mail forensic tools.

Cell phone and mobile device forensics: Understanding mobile device forensics, understanding acquisition procedures for cell phones and mobile devices.
UNIT - V

**Working with Windows and DOS Systems:** understanding file systems, exploring Microsoft File Structures, Examining NTFS disks, Understanding whole disk encryption, windows registry, Microsoft startup tasks, MS-DOS startup tasks, virtual machines.

**Textbooks:**

**References:**
1. Real Digital Forensics, Keith J. Jones, Richard Bejtich, Curtis W. Rose, Addison- Wesley Pearson Education
SOFTWARE METRICS AND MEASURES  
(Professional Elective – VI)

IV Year B.Tech. CSE II-Semester  

L    T    P    C  
3    0    0    3

Objectives  
1. Understand the basic techniques of data collection and how to apply them  
2. Learn software metrics that define relevant metrics in a rigorous way.

Outcomes  
1. Perform some simple statistical analysis relevant to software measurement data.  
2. Use from practical examples both the benefits and limitations of software metrics for quality control and assurance.

UNIT - I  
Measurement Theory  

UNIT - II  
Data Collection And Analysis  

UNIT - III  
Product Metrics  

UNIT - IV  
Quality Metrics  
Software quality metrics – Product quality – Process quality – Metrics for software maintenance – Case studies of Metrics Program – Motorola – HP and IBM.

UNIT - V  
Management Metrics  

Textbooks:  
2. Software Metrics ; A Rigorous approach Fenter Norman, E., Chapmen & Hall, London.

References:  