### III-Semester (2 Year)

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Course code</th>
<th>Course title</th>
<th>Hours Per Week</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>22PC3CS17</td>
<td>Discrete Mathematics</td>
<td>3 1 0</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>22PC3CS05</td>
<td>Object Oriented Programming using Java*</td>
<td>3 1 2</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>22PC3CS02</td>
<td>Database Management Systems*</td>
<td>3 0 2</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>22PC3CS03</td>
<td>Operating Systems</td>
<td>3 0 0</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>22PC3CS04</td>
<td>Operating Systems Lab</td>
<td>0 0 2</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>22PR3CS01</td>
<td>Evaluation of Internship-I</td>
<td>0 1 0</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>22PR3CS02</td>
<td>SSDC/Doing Engineering-I</td>
<td>0 1 2</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td><strong>TOTAL</strong></td>
<td></td>
<td>12 4 8</td>
<td>20</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Course code</th>
<th>Course title</th>
<th>Hours Per Week</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>22AC3HS01</td>
<td>Universal Human Values</td>
<td>0 2 0</td>
<td>0</td>
</tr>
</tbody>
</table>

### IV-Semester (2 Year)

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Course code</th>
<th>Course title</th>
<th>Hours Per Week</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>22BS4MT04</td>
<td>Probability and Statistics</td>
<td>3 1 0</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>22PC4CS11</td>
<td>Formal Language and Automata Theory</td>
<td>3 0 0</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>22PC4CS06</td>
<td>Design and Analysis of Algorithms*</td>
<td>3 0 2</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>22PC4CS13</td>
<td>Computer Organization and Architecture</td>
<td>3 0 0</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>22PC4CS07</td>
<td>Computer Networks*</td>
<td>3 0 2</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>22BS4EG03</td>
<td>English for Employability*</td>
<td>1 0 2</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td><strong>TOTAL</strong></td>
<td></td>
<td>16 1 6</td>
<td>20</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Course code</th>
<th>Course title</th>
<th>Hours Per Week</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>22MC4HS06</td>
<td>Constitution of India</td>
<td>0 1 0</td>
<td>0</td>
</tr>
</tbody>
</table>
Discrete Mathematics

Prerequisite: Knowledge of Simple Mathematics.

Course Objectives:
1. To introduce the mathematical fundamentals and construct direct and indirect method proofs.
2. To perform Relations with their properties and Apply counting principles to determine probabilities.
3. To introduce generating functions and recurrence relations.
4. Use different traversal methods for Trees and Graphs

Course Outcomes:
1. Understand the basic concepts of Mathematical reasoning and basic counting techniques. Also understand the different types of proves like mathematical induction.
2. Understand the concepts of various types of relations, partial ordering and equivalence relations.
3. Apply the concepts of generating functions to solve the recurrence relations and principle of inclusion and exclusion to solve some simple algorithms in discrete mathematics
4. Understand various definitions in graph theory and study their properties.

Module - I

Module - II
Relations and Their Properties: Representing Relations, Closure of Relations, Partial Ordering, Equivalence Relations, compatibility and partial ordering relations and Hasse diagram. Set Theory: Review of Sets and Functions
Module - III:
**Advanced Counting Techniques and Relations:** Recurrence Relations, Solving Recurrence Relations, Generating Functions, Solutions of Homogeneous Recurrence Relations, Divide and Conquer Relations, The Binomial and Multinomial Theorems, Inclusion-Exclusion.

Module - IV

**Number Theory:** Divisibility and Factorization. Congruences. Simultaneous linear congruences, Chinese Remainder Theorem. Wilson’s Theorem, Fermat’s Theorem, pseudoprimes and Carmichael numbers, Euler’s Theorem. Arithmetic functions and Quadratic residues.

Module - V:

**Graphs and Trees:** Graphs and Graph Models, Graph Terminology and Special Types of Graphs, Representing Graphs, Bipartite Graphs and Graph Isomorphism, Connectivity, Euler and Hamilton Paths, Planar Graphs, Graph Colouring.

**Trees:** Introduction to Trees, Applications of Trees, Spanning Trees, Minimum Spanning Trees.

**Text Books:**


**Reference Books:**


**CO-PO/PSO Mapping:**

<table>
<thead>
<tr>
<th>CO-PO</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>PSO 1</th>
<th>PSO2</th>
<th>PSO3</th>
</tr>
</thead>
<tbody>
<tr>
<td>CO-1</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO-2</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO-3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO-4</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Object Oriented Programming Using Java

Prerequisites: Programming in c.

Course Objectives:
1. Understand fundamental concepts and object oriented concepts in java
2. Implementing the concept of packages and exception handling in Java.
3. Implement the concept of multithreading and interprocess communication in java.
4. Develop GUI applications.

Course Outcomes:
1. Solve the given problem using OOPS technique.
2. Explain the concept of Package and Exception Handling.
3. Implement Multi threading and Inter process communication in java
4. Develop GUI based applications using applet, awt, Event handling and swing.

Module- I
History and Evolution of java: Java’s lineage, Java and internet, Byte code, Java buzzwords, Evolution of java.
Object oriented programming - data, types, variables, Arrays, operators, control statements, type conversion and casting, Introduction to classes, objects, methods, constructor, this and static keywords, garbage collection, overloading methods, parameter passing, access control, Command line arguments, exploring String class
Inheritance: member access and inheritance, Multilevel Inheritance, super and final keywords, method overriding, dynamic method dispatch, abstract classes and methods.

Programs:
1. A. Develop a java Program to find the roots of Quadratic equation.
   B. Develop a java Program to find the Fibonacci sequence.
2. A. Develop a java program to demonstrate the concept of method overloading.
   B. Develop a java program to demonstrate the concept of method overriding.
3. A. Develop a java program to demonstrate the concept of multi level inheritance.
   B. Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named print Area (). Provide three classes named Rectangle, Triangle, and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method print Area () that prints the area of the given shape.
Module- II

Packages and Interfaces: Defining, Creating and Accessing a Package, understanding CLASSPATH, importing packages, Differences between classes and interfaces, defining an interface, implementing interface, applying interfaces, variables in interface and extending interfaces.

Exception handling: Concepts of exception handling and its benefits, usage of try, catch, throw, throws and finally, built in exceptions, creating own exceptions.

Programs:
1. A. Develop a java program to demonstrate the significance of multiple catch.
   B. Develop a java program to demonstrate throws clause.

2. Develop a Java program using packages to demonstrate access control modifiers.

Module- III

Multithreading: Differences between multi-threading and multi programming, thread life cycle, creating threads using thread class and Runnable interface, thread priorities, synchronization , interthread communication.

I/O Streams: Stream classes, Byte and character streams, File class, reading and writing files, reading and writing from console, serialization.

Programs:
1. Develop a Java program that implements a multithread application that has three threads. First thread generates random integer every 1 second and if the value is even, second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of cube of the number.
2. Develop a Java program that correctly implements producer consumer problem using the concept of inter thread communication.
3. A. Develop a java program that copies the content of one file to another.
   B. Develop a java Program to accept data from keyboard & develop it into a file.

Module- IV

Applets: Concepts of Applets, differences between applets and applications, life cycle of anapplet, creating applets, passing parameters to applets.

AWT: class hierarchy, user interface components- labels, buttons, scrollbars, text components, checkbox, checkbox groups, choices, lists panels – scroll pane, dialogs, menu bar, Layout Managers- Flow Layout, Border Layout, Grid Layout, Card Layout, Grid Bag Layout.

Event Handling: Events, Event sources, Event classes, Event Listeners, Delegation event model, handling mouse and keyboard events, Adapter classes.

Programs:
1. A. Develop an applet in Java that receives an integer in one text field, and computes its factorial value and returns it in another text field, when the button named “Compute” is clicked.
   B. Develop a java program for passing parameters to applets
2. Develop a Java program to demonstrate Mouse Listener, Mouse Motion Listener
3. Develop a Java program that simulates a traffic light. The program lets the user select one of three lights: red, yellow, or green with radio buttons. On selecting a button, an appropriate message with “Stop” or “Ready” or “Go” should appear above the buttons in selected color. Initially, there is no message shown.

Module –V


Programs:

1. Develop a java Program to design a calculator for implementing basic functions like +,*,-, / using grid layout.
2. Develop an applet that moves the character up, down, left and right when the appropriate arrows are pressed.

Text Books:

Reference Books:
1. An Introduction to JAVA Programming (Chapter 6) Author: Y.Daniel Liang , Publishers:Tata Mec-Hill.
3. Internet and Java Programming, R.KrishnaMurthu and S.Prabhu, New Age Publishers
## CO-PO/PSO Mapping:

<table>
<thead>
<tr>
<th>Course Outcome(s)(COs)</th>
<th>Program Outcomes (POs)</th>
<th>Program Specific Outcomes*</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PO 1 PO 2 PO 3 PO 4 PO 5 PO 6 PO 7 PO 8 PO 9 PO 10 PO 11 PO 12</td>
<td>PSO 1 PSO 2 PSO 3</td>
</tr>
<tr>
<td>CO1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>CO2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>CO3</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>CO4</td>
<td>3 1 1 1 1 3</td>
<td></td>
</tr>
</tbody>
</table>
Database Management Systems

Prerequisite: Basics of computer programming language, Data structures.

Course Objectives:
1. To enable students, define and describe basic concepts of Relational database managements and applications.
2. To provide students the theoretical concepts of data models and database design and normal forms.
3. To make students familiarize with relational model, relational algebra, transaction control and concurrency control.
4. To master the basics of SQL, PL/SQL and design queries.
5. To introduce storage structures and access techniques.

Course Outcomes:
1. Describe fundamentals of RDBMS, database design and normal forms.
2. Design SQL & PL/SQL for retrieval and management of data.
3. Understand basics of transaction processing and concurrency control.
4. Summarize database storage structures and access techniques.

Module- I

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

Introduction to Database Design: Database Design and ER Diagrams, Entities, Attributes, and Entity Sets, Relationships and Relationship Sets, Super key, candidate key, Participating constraints, Weak entity, Additional Features of the ER Model, Conceptual Design with the ER Model.

Programs:
1. E-R Model:
   Analyze any problem carefully and come up with the entities in it. Identify what data has to be persisted in the database. This contains the entities, attributes etc. Identify the primary keys for all the entities. Identify the other keys like candidate keys, partial keys, if any.
2. Concept Design with E-R Model
   Relate the entities appropriately . Apply cardinalities for each relationship. Identify strong entities and weak entities (if any). Indicate the type of relationships (total / partial). Try to incorporate generalization, aggregation, specialization etc where ever required.
3. Installation of Mysql / SQL for practicing commands
Module- II

SQL: Introduction To SQL, Query Languages, Basic SQL Query. Introduction to views, destroying/altering tables and views. Joins.

Relational Algebra and Calculus: Selection and Projection, Set operations, Joins, Division, More examples on Algebra queries, Tuple relational Calculus, Domain Relational Calculus.

Programs:
1. Practicing DDL commands
2. Practicing DML commands
3. Querying (using ANY, ALL, IN, Exists, NOT EXISTS, UNION, INTERSECT, Constraints etc.)
4. Views, joins.

Module- III

Advanced SQL: SQL Functions, Aggregate Operators, Group by & having clause, Sub queries, Nested Queries, triggers and active data bases, cursors, procedures.

Schema Refinement: Problems caused by redundancy, decompositions, problems related to decomposition, reasoning about functional dependencies, 1NF, 2NF, 3NF, 3.5NF, lossless join decomposition, multi-valued dependencies, 4NF & 5NF.

Programs:
1. Queries using Aggregate functions, Group by & Having.
2. Apply Normalization(1NF, 2NF, 3NF, 4NF & 5NF)
3. Triggers (Creation of insert trigger, delete trigger, update trigger)
4. Procedures
5. Usage of Cursors

Module -IV

Transaction Management: The ACID Properties, Transactions and Schedules, Concurrent Execution of Transactions, 2PL, Serializability, Time stamp based protocol, validation based Protocol. Implementation of isolation, Multiple granularity,

Recoverability: Recoverability, Introduction to Lock Management, Lock Conversions, Dealing with Deadlocks, shadow paging.

Programs:
1. Practicing DCL commands
2. Practicing TCL commands

Module -V

Storage and Indexing: Data on External Storage, File Organization and Indexing, Cluster Indexes, Primary and Secondary Indexes, Index data Structures, Hash Based Indexing, Tree base Indexing.

Tree Structured Indexing: Intuitions for tree Indexes, Indexed Sequential Access Methods (ISAM), B+ Trees.

Programs:
1. Apply indexing methods (primary, secondary, hashing, spars & dense indexing)

   Example: An organization contains several employees in each department. Suppose we use a clustering index, where all employees which belong to the same Dept_ID are
considered within a single cluster, and index pointers point to the cluster as a whole. Here Dept_Id is a non-unique key.

2. Write PL/SQL program for B-tree
3. Write PL/SQL program for B+tree

Text Books:


Reference Books:


Software’s Required:
Programs are to be developed using My SQL / SQL / PL/SQL.

CO-PO/PSO Mapping:

<table>
<thead>
<tr>
<th>Course Outcomes (COs)</th>
<th>Program Outcomes (POs)</th>
<th>Program Specific Outcomes*</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PO 1</td>
<td>PO 2</td>
</tr>
<tr>
<td>CO1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CO2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CO3</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>CO4</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>
Operating Systems

Course Objective:
1. To understand the components, operations of the operating system
2. To interpret the scheduling policies and memory management issues
3. To understand the process concurrency and synchronization
4. To understand the concept of file management

Course Outcomes:
At the end of the course student will be able to
1. Understand the structure of Operating System and its architecture
2. Apply the scheduling strategies for real time implementations
3. Illustrate synchronization problems, deadlock and its techniques
4. Apply Memory Management Techniques

Module - I


Module - II


Module- III

CPU Scheduling- Scheduling algorithms, Multiple-Processor Scheduling, Real-Time Scheduling, Threads- Overview, Multithreading models, threading issues.

Process Coordination – Process Synchronization, Theoretical Section Problem, Peterson's solution, Synchronization Hardware, Semaphores, and Classic Problems of Synchronization,
Monitors, Case Studies: Linux, Windows.

**Module - IV**


**Memory Management**: Memory address, Swapping and Managing Free Memory Space, Resident Monitor, Multiprogramming with Fixed Partition, Multiprogramming With Variable Partition, Multiple Base Register,

**Module- V**


**TEXT BOOKS:**


**REFERENCE BOOKS:**


**CO-PO &PSO Mapping:**

<table>
<thead>
<tr>
<th>Course Name - Course Outcomes / Program Outcomes</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>PSO 1</th>
<th>PSO 2</th>
<th>PSO 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>CO1</td>
<td>M</td>
<td>L</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO2</td>
<td>H</td>
<td>H</td>
<td>M</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO3</td>
<td>M</td>
<td>H</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO4</td>
<td>M</td>
<td>L</td>
<td>L</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Operating Systems Lab

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

Course 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, inter process communication and I/O in Unix.

Course 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    Round Robin    d) priority
2. Write a C program to simulate Bankers Algorithm for Deadlock Avoidance and Prevention.
3. Write a C program to implement the Producer – Consumer problem using semaphores.
4. Write a C program to simulate the concept of Dining-philosophers problem.
5. Write C programs to simulate the following memory management techniques
   a) Paging    b) Segmentation
6. Write C programs to illustrate the following IPC mechanisms
   a) Pipes    b) FIFOs    c) Message Queues    d) Shared Memory
7. Write a C program to simulate the following contiguous memory allocation Techniques
   a) Worst fit    b) Best fit    c) First fit
8. Simulate all File Organization Techniques
   a) Single level directory          b) Two level directory

9. Write a C program to simulate the following contiguous memory allocation Techniques
   a) Worst fit         b) Best fit            c) First fit.

10. Implementation of the following Page Replacement Algorithms
    a) FIFO           b) LRU              c) LFU

TEXT BOOKS:


REFERENCE BOOKS:


CO-PO & PSO Mapping:

<table>
<thead>
<tr>
<th>Course Name - Course Outcomes / Program Outcomes</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>0</th>
<th>11</th>
<th>12</th>
<th>PSO 1</th>
<th>PSO 2</th>
<th>PSO 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>CO1</td>
<td>M</td>
<td>L</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO2</td>
<td>H</td>
<td>H</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>M</td>
</tr>
<tr>
<td>CO3</td>
<td>M</td>
<td>H</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO4</td>
<td>M</td>
<td>L</td>
<td>L</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Course Objective:
The goals and objectives of engineers revolve around creating processes and designs, as well as maintenance and operations duties in a variety of different disciplines.

Course Outcomes:
At the end of the course student will be able to

- Able to understand the controllers working.
- Analysis the cloud based projects
- Understand that real time monitoring system.
- Create a cloud Account And database system.

Module 1
Microprocessor and Microcontroller
Introduction to Microprocessor, architecture of Microprocessor, Introduction to Microcontroller, Architecture of microcontroller, difference between microprocessor and microcontroller.

Module 2
Arduino and ESP32
Introduction to arduino, types of arduino boards, pin description of arduino board, Introduction to Esp32, pin description of Esp32 board, Programming and steps for installation.

Module 3
Cloud computing
Introduction, types of services, types of deployment models, Edge Computing, fog computing, working and uses of cloud computing, Advantages of cloud computing.

Module 4
AWS Cloud computing
Introduction to AWS, Moving to the AWS cloud, AWS Global Infrastructure, AWS Services, Amazon VPC, VPC networking and security.
Module 5

Compute

Compute Service overview, Amazon EC2, AWS EBS, Working with EBS, Amazon RDS, build a database server, AWS well architected framework design principles.

Tasks:

I) Autonomous vehicles using Edge computing

II) In-hospital patient monitoring

III) Building IoT monitoring with cloud technology

IV) IoT Based Solar Power Monitoring System with ESP32

V) Telegram bot with ESP32- Control GPIO pins through telegram chat.

VI) ESP32 GPS Tracker- IoT based Vehicle Tracking System

VII) AWS IoT with arduino ESP32.

TEXT BOOKS:


REFERENCE BOOKS:

Universal Human Values (UHV)

(Common to ECE, EEE, ME, CSE, CSD, CSO, CSC, CSM branches)

Course Objectives:

1. To help students distinguish between values and skills, and understand the need, basic guidelines, content and process of value education.
2. To help students initiate a process of dialog within themselves to know what they ‘really want to be’ in their life and profession
3. To help students understand the meaning of happiness and prosperity for a human being.
4. To facilitate the students to understand harmony at all the levels of human living, and live accordingly.
5. To facilitate the students in applying the understanding of harmony in existence in their profession and lead an ethical life

Module I

UNIT 1: Self-Exploration on UHV Basic Guidelines

Content and Process for Value Education Understanding the need, basic guidelines, Self-Exploration–what is it? - its content and process; ‘Natural Acceptance’ and Experiential Validation- as the mechanism for self-exploration.

UNIT 2: Continuous Happiness and Prosperity

A look at basic Human Aspirations, Right understanding, Relationship and Physical Facilities- the basic requirements for fulfilment of aspirations of every human being with their correct priority, Understanding Happiness and Prosperity.

Module II

UNIT 1: Understanding Harmony in the Human Being

Harmony in Myself Understanding human being as a co-existence of the sentient ‘I’ and the material ‘Body’, Understanding the needs of Self (‘I’) and ‘Body’ - Sukh and Suvidha, Understanding the Body as an instrument of ‘I’

UNIT 2: Understanding Harmony in self

Understanding the characteristics and activities of ‘I’ and harmony in ‘I’, Understanding the harmony of I with the Body: Sanyam and Swasthya; correct appraisal of Physical needs, meaning of Prosperity in detail,
Module III

UNIT 1: Understanding Harmony in the Family
Harmony and Values in Relationships in the Family- the basic unit of human interaction, Trust (Vishwas) and Respect (Samman) as the foundational values of relationship, Understanding the meaning of Vishwas & Samman; Difference between intention and competence,

UNIT 2: Understanding Harmony in the Society
Understanding the harmony in the society: Samadhan, Samridhi, Abhay, Sah-astitva as comprehensive Human Goals, Visualizing Undivided Society (Akhand Samaj), Universal Order (SarvabhaumVyawastha).

Module IV

UNIT 1: Understanding Harmony in the Nature and Existence
Whole existence as Co-existence Understanding the harmony in the Nature, Interconnectedness and mutual fulfilment among the four orders of nature- recyclability and self-regulation in nature

UNIT 2: Understanding Harmony in the Existence
Understanding Existence as Co-existence (Sah-astitva) of mutually interacting units in all-pervasive space, Holistic perception of harmony at all levels of existence.

Module V

UNIT 1: Exploring Attitudes towards gender

UNIT 2: Gender relationship and Culture
Gender roles and relationship matrix, sex selection and consequences, declining sex ratio, Gender Issues- Gender sensitive language, Just Relationships: Being together as equals.

Text Books:

2. Towards a World of Equals: a bilingual Textbook on Gender. A Suneetha, and others... Telugu Academy, Telangana Gov. 2015
References:


MOOC Course: NPTEL - Exploring Human Values: Visions of Happiness and Perfect Society - Web course

Course Outcome: On completion of this course, the students will be able to

CO1: Explore on the basic aspiration of Human being and its fulfilment

CO2: Distinguish the difference between the Self and the Body

CO3: Explore the value of harmony in family, society and nature

CO4: Understanding of gender related issues and gender relationship.

<table>
<thead>
<tr>
<th>Course Outcomes (COs)</th>
<th>Program Outcomes (POs)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PO 1</td>
</tr>
<tr>
<td>CO1</td>
<td></td>
</tr>
<tr>
<td>CO2</td>
<td></td>
</tr>
<tr>
<td>CO3</td>
<td></td>
</tr>
<tr>
<td>CO4</td>
<td></td>
</tr>
<tr>
<td>CO5</td>
<td></td>
</tr>
</tbody>
</table>

O-PO Mapping Chart
(3/2/1 indicates strength of correlation) 3 – High; 2 – Medium; 1 – Low
PROBABILITY AND STATISTICS
(CSE)

Pre-requisite: Basic knowledge of set and relations theory, permutations, combinations, Venn diagrams, measures of central tendency and dispersion.

Course Objectives: To provide the student with
1. The theory of Probability
2. Probability distributions of single and multiple random variables.
3. The sampling theory, point estimation and interval estimation
4. Testing of hypothesis and making statistical inferences.
5. Concept of Correlation and Regression.

MODULE I

MODULE II
Discrete Probability Distributions: Binomial Distribution, Poisson distribution and statistical constants of these distributions using moment generating function.
Continuous Probability Distributions: Uniform Distribution, Exponential Distribution, mean and variance of these distributions using moment generating function, Normal Distribution and its related applications.

MODULE III
Sampling Distribution: Random Sampling, Some Important Statistics, Sampling Distributions, Sampling Distribution of Means, variance and the Central Limit Theorem.
MODULE IV
Tests of Significance (Large Samples): test for single mean, difference of means, single proportion, difference of proportions.
Tests Of Significance (Small Samples): t-Test for single mean, difference of means, paired t-test, F-test, Chi-square test for goodness of fit and independence of attributes.

MODULE V
Bivariate Distribution: Joint Probability distributions - Joint Probability mass function, joint probability density function, Marginal Distribution, Covariance of two random variables.
Correlation And Regression: Karl Pearson coefficient of correlation, Rank correlation, Regression coefficient, Lines of regression.

Text Books:

Reference Books:

MOOC courses:
1. Probability: http://nptel.ac.in/courses/111105041/
2. Probability and Statistics: http://nptel.ac.in/courses/111105035/
3. Probability: https://nptel.ac.in/courses/111/102/111102111/

E- Books:
**Course Outcomes:** After learning the contents of this paper the student must be able to

CO1: Compute probabilities using theorems in probability.
CO2: Solve problems involving univariate and bivariate random variables of probability distributions.
CO3: Apply Inferential Statistics to make predictions or judgments about the population from which the sample data is drawn for large samples and small samples.
CO4: Establish relationships between variables using correlation and regression.

<table>
<thead>
<tr>
<th>Course Outcomes (COs)</th>
<th>Program Outcomes (POs)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PO 1</td>
</tr>
<tr>
<td>CO1</td>
<td>3</td>
</tr>
<tr>
<td>CO2</td>
<td>3</td>
</tr>
<tr>
<td>CO3</td>
<td>3</td>
</tr>
<tr>
<td>CO4</td>
<td>3</td>
</tr>
</tbody>
</table>

**CO-PO/PSO Mapping Chart**
(3/2/1 indicates strength of correlation)
3 – High; 2 – Medium; 1 – Low
Formal Languages and Automata Theory

Prerequisite: Discrete Mathematical Structures knowledge

Course Objectives:

1. Determine the relationship between languages and machines and understand their power
2. Explain deterministic and non-deterministic machines
3. Explain the representation of Regular expressions
4. Understand the decidability and undecidability of problems

Course Outcomes:

1. Design Finite Automata for the given language
2. Write Regular expression for programming language constructs
3. Design Context free grammars for formal languages
4. Design Turing Machine and check for the decidability and undecidability of the language

Module - I

Introduction: Structural Representations, Automata and Complexity, the Central Concepts of Automata Theory – Alphabets, Strings, Languages, Deterministic Finite Automata (DFA), Non-deterministic finite state automata.

Equivalence of NFA: Equivalence with DFA, NFA with $\varepsilon$ - moves, Conversion to NFA without $\varepsilon$ - moves, minimization of finite automata, equivalence between FAs, Finite Automata with Outputs – Mealy machine, Moore machine.

Module - II

Regular Expression: Finite Automata and Regular Expressions, Applications of FA, Properties of regular sets, Conversion of Finite Automata to Regular Expressions. Pumping Lemma, Closure properties of Regular Expressions


Module - III

Pushdown Automata—Definitions, The languages of PDA, Equivalence of PDAs and CFGs, Deterministic Pushdown Automata. From CFG to PDA, From PDA to CFG.

Module - IV
Turing Machines: Introduction to Turing Machine, Formal Description, Instantaneous description, Techniques for construction of Turing machines. Modifications of TM.
Types of Turing machine: Turing machines and halting

Module - V

Text Books:

Reference Books:

Web Resources:
1. JFLAP Simulator download link: http://www.jflap.org/
3. https://nptel.ac.in/courses/106/103/106103070/

CO-PO/PSO Mapping:

<table>
<thead>
<tr>
<th>CO-PO</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>PSO1</th>
<th>PSO2</th>
<th>PSO3</th>
</tr>
</thead>
<tbody>
<tr>
<td>CO-1</td>
<td></td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>CO-2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO-3</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO-4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Prerequisites: 
Data Structure, Discrete Mathematics

Course Objectives
1. To analyze the performance of algorithms.
2. To choose the appropriate data structure & algorithm design method for specific application
3. To understand how the choice of data structure & algorithm design method impact the performance of program
4. To design efficient algorithms for different problems

Course Outcomes:
At the end of the Course the Students will be able to
1. Describe computational solution to well-known problems like searching, sorting etc
2. Estimate the computational complexity of different algorithms.
3. Apply different designing methods for development of algorithms to realistic problems through greedy, dynamic programming, back tracking.
4. Devise an algorithm using appropriate design strategies for problem solving

Module - I
Introduction: Algorithm, Pseudo code for expressing algorithms, Performance Analysis-Space complexity, Time complexity, Asymptotic Notation- Big oh notation, Omega notation, Theta notation and Little oh notation, Probabilistic analysis, Amortized Complexity.
Divide and conquer: General method, applications-Binary search, Quick sort, Merge sort, Strassen’s matrix multiplication

Programs:
1. Sort a given set of elements using the quick sort method and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the 1st to be sorted and plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator.
2. Implement merge sort algorithm to sort a given set of elements and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator.
HR22

Module – II

Searching and Traversal Techniques: Efficient non-recursive binary tree traversal algorithms, disjoint set operations, union and find algorithms, spanning trees, Graph traversals-Breadth First Search and Depth First Search, AND/OR Graphs, game tree, connected components and biconnected components.

Programs:

1. Write a program, from a given vertex in a weighted connected graph, find shortest
   i. paths to other vertices using Dijkstra’s algorithm.
2. Write a C program to implement the Stack using arrays. Write Push(), Pop(), and
   i. Display() methods to demonstrate its working.

Module – III

Backtracking: General method, applications-n-queen problem, sum of subsets problem, graph coloring, Hamiltonian cycles.
Greedy method: General method, applications-Job sequencing with deadlines, 0/1 knapsack problem, Minimum cost spanning trees: Single source shortest path problem.

Programs:

1. Design and implement in Java to find all Hamiltonian Cycles in a connected undirected Graph G of n vertices using backtracking principle.
2. Find Minimum Cost Spanning Tree of a given connected undirected graph using Prim's algorithm

Module – IV

Dynamic Programming: General method, applications- 0/1 knapsack problem, All pairs shortest path problem: Travelling sales person problem, Reliability design.

Programs:

1. Implement a C Program to implement the 0/1 Knapsack problem using (a) Dynamic Programming method (b) Greedy method.
2. Write C programs to (a) Implement All-Pairs Shortest Paths problem using Floyd's algorithm. (b) Implement Travelling Sales Person problem using Dynamic programming
Module – 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, NP-hard problems.

Programs
1. Design an algorithm and implement a program to find a subset of a given set $S = \{S_1, S_2,..,S_n\}$ of $n$ positive integers whose SUM is equal to a given positive integer $d$. For example, if $S = \{1, 2, 5, 6, 8\}$ and $d= 9$, there are two solutions $\{1,2,6\}$and $\{1,8\}$. Display a suitable message, if the given problem instance doesn't have a solution.

Text Books:

Reference Books:

CO-PO/PSO Mapping:

<table>
<thead>
<tr>
<th>COs</th>
<th>CO1</th>
<th>CO2</th>
<th>CO3</th>
<th>CO4</th>
</tr>
</thead>
<tbody>
<tr>
<td>PO1</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td></td>
<td>O2</td>
<td>O3</td>
<td>O4</td>
<td>O5</td>
</tr>
<tr>
<td>P0</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>P1</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>P2</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>P3</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>P4</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>
Course Objectives:

1. To familiarize student with computer architecture and organization.
2. Prepare student to perform mathematical operation and execute complete instruction in computer.
3. Prepare students to analyze performance of various memories.
4. To familiarize student with input/output operation and interrupt handling mechanism.

Course Outcomes:

1. On Successful completion of course, students will be able to:
2. Describe basic components of a computer, including CPU, memories, and input/output, and their organization.
3. Execute complete instruction and design control unit.
4. Perform mathematical operations on arithmetic and floating point numbers.
5. Analyse cost performance trade off in designing memory hierarchy and instruction sets.

Module - I

Basic Structure of Computers: Functional units of computer. Instructions set architecture of a CPU-Instruction sequencing, addressing modes, and instruction set classification, subroutine & parameter passing, expanding opcode, RISC and CISC.

Module – II

Basic Processing Unit: Bus architecture, Execution of a Complete Instruction, sequencing of control signals, Hardwired control, Micro-programmed Control.

Module – III

Data Representation: signed number representations and their operations, Computer arithmetic – integer addition and subtraction, design of Fast Adders, Multiplication- shift and add, booth's Algorithm, bit-pair recoding, Integer Division- restoring and non-restoring division. Floating point numbers- representation, arithmetic, guard bits and rounding.

Module – IV
Concept of hierarchical memory, Memory System Design: Semiconductor RAM memories, Static and Dynamic Memories, ROM, higher order memory design, multi-module memories, Memory interleaving, Cache memory, Cache size vs. block size, mapping functions, replacement algorithms, Cache read/write policy, Virtual Memory. Secondary storage – Magnetic disk, Optical disk.

Module – V

Input/output Organization: I/O mapped I/O and memories mapped I/O, interrupt and interrupt handling mechanisms, vectored interrupts, synchronous vs. asynchronous data transfer, Bus Arbitration, Direct Memory Access,

Pipelining: Basic concepts of pipelining, throughput and speedup, Introduction of Parallel Computing: SISD, MISD, SIMD, MIMD

Text Books:
2. W. Stallings; Computer Organization & Architecture; PHI publication; 2001
3. M Mano; Computer System and Architecture; PHI publication; 1993.

Reference Books:
1. A.S.Tanenbaum; Structured Computer Organization; Prentice Hall of India Ltd.
2. J. P. Hayes; Computer Architecture & Organization; 3rdedition; McGraw-Hill; 1998

CO-PO/PSO Mapping:

<table>
<thead>
<tr>
<th>Course Outcomes (COs)</th>
<th>Program Outcomes (POs)</th>
<th>Program Specific Outcomes*</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PO 1</td>
<td>PO 2</td>
</tr>
<tr>
<td>CO1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>CO2</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>CO3</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>CO4</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>
Computer Networks

Course Objectives:
1. To understand the concept of layering in networks.
2. To know the functions of protocols of each layer of TCP/IP protocol suite.
3. To visualize the end-to-end flow of information.
4. To understand the components required to build different types of networks.
5. To learn concepts related to network addressing and routing.

Course Outcomes:
On the completion of the course, the student will be able to:
1. Identify the devices and protocols to design a network and implement it.
2. Build network applications using the right set of protocols and estimate their performances.
3. Apply addressing principles such as subnetting and VLSM for efficient routing.
4. Explain media access and communication techniques.

Module – I

Programs:
1. Implement the data link layer framing methods such as character, character-stuffing and bit stuffing.
2. 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

Module - II

Programs:
1. Write a program to compute CRC code for the polynomials CRC-12, CRC-16 and CRC CCIP
2. 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.
3. Write a program for congestion control using Leaky bucket algorithm.

Module – III


Programs:
1. Take an example subnet of hosts and obtain a broadcast tree for the subnet.
2. Write a program for frame sorting technique used in buffers.

Module - IV

Programs:
1. Implement Dijkstra’s algorithm to compute the shortest path through a network
2. Implement distance vector routing algorithm for obtaining routing tables at each node.

Module - V:

Programs:
2. 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.

Text Books:


References Books:

## CO and PO Mapping

<table>
<thead>
<tr>
<th></th>
<th>PO1</th>
<th>PO2</th>
<th>PO3</th>
<th>PO4</th>
<th>PO5</th>
<th>PO6</th>
<th>PO7</th>
<th>PO8</th>
<th>PO9</th>
<th>PO10</th>
<th>PO11</th>
<th>PO12</th>
</tr>
</thead>
<tbody>
<tr>
<td>CO1</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO2</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO3</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td></td>
<td></td>
<td></td>
<td>√</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO4</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Course Objectives:
1. To improve the communication skills, body language, facial expression and gesture.
2. To be able to understand the concept of employability skills (Quantum dexterity) and enhancing ones’ behavior in the personal, professional and social forum.
3. To evaluate the LSRW (listening, speaking, reading and writing) through assessment.
4. To learn the basic grammar for improving spoken and written communication.
5. To become problem solver, analyze and apply critical and analytical skills.
6. To Identify the Employability skills, assigning tasks (Group Discussion, JAM, Role play etc.,) for day today evaluation.

Course Outcomes: After undergoing this course, the student will be able to;
1. Understand the explicit and implicit of importance of employability skills.
2. Demonstrate life skills like team work, learning skills, problem solving, attitude, adaptability and flexibility.
3. Apply critical and analytical skills to bring out the solution on problem/case study.
4. Recognize the need of appropriate words, Phrases & functional grammar and apply them in both spoken and written communication.

Module I: Wings of Fire
“Orientation” an extract from Wings of fire- An Autobiography of Abdul Kalam by Arun Tiwari.
Grammar
Vocabulary- Omission of Articles, Collective Nouns, Prepositions, Collocations.
Reading- Observation Passage, Survey Passage
Writing- Narrative & Descriptive writing.

Module II: 5 Points Someone
The Gift an extract from 5 Points Someone by Chetan Bhagat
Grammar -
Vocabulary- Advanced Collocations, Proverbs, Idioms, One word Substitute
Reading – Complex passage, Reading Comparison,
Writing- Usage of Idioms and Proverbs in Passage

Module III: Wise Leaders Wanted & Shift Your Perspective: Connect to Your Noble Purpose
“Wise Leaders Wanted & Shift Your Perspective: Connect to Your Noble Purpose” an extract from From Smart to Wise: Acting and Leading with Wisdom Kaipa, Prasad, and Navi Radjou.
Grammar
Vocabulary- Technical vocabulary, Auxiliaries and Modals,
Reading- Technical Comprehension,
Writing- Creative Resume.

Module IV: Variation Under Nature
“Variation Under Nature” an extract from Origin of Species by Charles Darwin

Grammar
Vocabulary- Coherence-Cohesive devices, Figures of speech
Reading- Inferring Reading, Reciting and Reviewing (SQ3R)

Module V: Let's Build a Company: A Start-up Story Minus the Bullshit
Let's Build a Company: A Start-up Story Minus the Bullshit by Harpreet Grover and Vibhore Goyal

Grammar
Vocabulary- Topic/Situation based Vocabulary, Tongue Twisters.
Reading- Critical Reading of known/unknown passages
Writing- Common Errors in Tenses, Description of hobbies, Future plans,
Reporting Speech: Direct & Indirect Speech, Email Writing, Formal letter writing (Enquiry, Apology, Leave, Request) Notice Writing, Information Transfer, Technical report writing

ACTIVITY IN LABS

Activity 1:
Narration (Historical places, events, Picture narration, Memorable incidents of life)
Self Intro, Daily Routine, Likes & Dislikes, Vocabulary, Triangular Activity (Person based- S-P), Imperatives & JAM
Targeted Skills- Listening- Speaking- Audio-Video clips

Activity 2: Quantum of Dexterity (QOD)
Ability (Personal, Behavioural & Professional) Request/Permission/Order, Survival kit, Career Objective Professional, Hidden Talents (Personal), Character Traits (Behavioural)
Targeted Skills- Reading-Writing – Concluding an open-ended Story, Creative Writing.

Activity 3: Critical & Analytical Skills
SWOC- (Social & Cultural, Political, Economic, Legal Impact, Technical, Nuances of Pronunciation, Voice Modulation, Neutralizing Mother Tongue Interference, Tongue Twisters for practice,
Targeted Skills- Writing SWOC, Self-Introduction, Exposure to a structured talk.

Activity 4: Flick Flow/Extempore
**Targeted Skills-** Speaking Skills

**Activity 5: On Job Training**
Formal & Informal communication, Resume E-mail Etiquette, Telephonic & Interview Etiquette, Situation based- Santa’s Bag, topic/case study-based Group Discussion, Kicks me! (Job Consultancy/Role Play)

**Targeted Skills-** Listening-Writing- Speaking

**Text Books**

**References**

**CO-PO MAPPING:**

<table>
<thead>
<tr>
<th>Course Outcomes (COs)</th>
<th>Program Outcomes (POs)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PO 1</td>
</tr>
<tr>
<td>CO1</td>
<td></td>
</tr>
<tr>
<td>CO2</td>
<td></td>
</tr>
<tr>
<td>CO3</td>
<td></td>
</tr>
<tr>
<td>CO4</td>
<td></td>
</tr>
</tbody>
</table>

*CO-PO Mapping Chart (3/2/1 indicates strength of correlation) 3 – High; 2 – Medium; 1 – Low*
The Constitution of India is the supreme law of India. Parliament of India cannot make any law which violates the Fundamental Rights enumerated under the Part III of the Constitution. The Parliament of India has been empowered to amend the Constitution under Article 368; however, it cannot use this power to change the “basic structure” of the constitution, which has been ruled and explained by the Supreme Court of India in its historical judgments. The Constitution of India reflects the idea of “Constitutionalism” – a modern and progressive concept historically developed by the thinkers of “liberalism” – an ideology which has been recognized as one of the most popular political ideology and result of historical struggles against arbitrary use of sovereign power by state. The historic revolutions in France, England, America and particularly European Renaissance and Reformation movement have resulted into progressive legal reforms in the form of “constitutionalism” in many countries. The Constitution of India was made by borrowing models and principles from many countries including United Kingdom and America.

The Constitution of India is not only a legal document but it also reflects social, political and economic perspectives of the Indian Society. It reflects India’s legacy of “diversity”. It has been said that Indian constitution reflects ideals of its freedom movement; however, few critics have argued that it does not truly incorporate our own ancient legal heritage and cultural values. No law can be “static” and therefore the Constitution of India has also been amended more than one hundred times. These amendments reflect political, social and economic developments since the year 1950. The Indian judiciary and particularly the Supreme Court of India has played an historic role as the guardian of people. It has been protecting not only basic ideals of the Constitution but also strengthened the same through progressive interpretations of the text of the Constitution. The judicial activism of the Supreme Court of India and its historic contributions has been recognized throughout the world and it gradually made it “as one of the strongest court in the world”.

**Course content**

1. Meaning of the constitution law and constitutionalism
2. Historical perspective of the Constitution of India
3. Salient features and characteristics of the Constitution of India
4. Scheme of the fundamental rights
5. The scheme of the Fundamental Duties and its legal status
6. The Directive Principles of State Policy – Its importance and implementation
7. Federal structure and distribution of legislative and financial powers between the Union and the States
8. Parliamentary Form of Government in India – The constitution powers and status of the President of India
9. Amendment of the Constitutional Powers and Procedure
10. The historical perspectives of the constitutional amendments in India
12. Local Self Government – Constitutional Scheme in India
13. Scheme of the Fundamental Right to Equality
14. Scheme of the Fundamental Right to certain Freedom under Article 19
15. Scope of the Right to Life and Personal Liberty under Article 21