<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></td>
<td></td>
<td></td>
<td>L</td>
<td>T</td>
</tr>
<tr>
<td>1</td>
<td>22BS3MT03</td>
<td>Statistical and Mathematical Foundations</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>22PC3CD01</td>
<td>Data Structures and Algorithms*</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>22PC3CS02</td>
<td>Database Management Systems*</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>22PC3CO01</td>
<td>Analog and Digital Electronics*</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>22PR3CO01</td>
<td>Evaluation of Internship-I</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>22PR3CO02</td>
<td>Student Skill Development Course/Doing Engineering-I</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>TOTAL</strong></td>
<td><strong>13</strong></td>
<td><strong>3</strong></td>
</tr>
<tr>
<td>7</td>
<td>22AC3HS01</td>
<td>Universal Human Values</td>
<td>0</td>
<td>2</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></td>
<td></td>
<td></td>
<td>L</td>
<td>T</td>
</tr>
<tr>
<td>1</td>
<td>22PC4CO02</td>
<td>Computer Organization and Architecture</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>22PC4CS06</td>
<td>Design and Analysis of Algorithms*</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>22PC4CO03</td>
<td>Sensors and Devices*</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>22PC4CS03</td>
<td>Operating Systems</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>22PC4CS07</td>
<td>Computer Networks*</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>22BS4EG03</td>
<td>English for Employability*</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>TOTAL</strong></td>
<td><strong>15</strong></td>
<td><strong>0</strong></td>
</tr>
<tr>
<td>7</td>
<td>22MC4HS06</td>
<td>Constitution of India</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
B. Tech II Year–I Sem

Subject Code: 22BS3MT03

L T P C
3 1 0 4

STATISTICAL AND MATHEMATICAL FOUNDATIONS
(Common to CSC/CSD/CSM/CSO)

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, Probability distributions of single and multiple random variables.
2. The sampling theory, point estimation and interval estimation.
3. Testing of hypothesis and making statistical inferences.
4. The Number Theory basic concepts useful for cryptography etc.

MODULE I

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

MODULE V
Greatest Common Divisors and Prime Factorization: Greatest common divisors, The Euclidean algorithm, The fundamental theorem of arithmetic, Factorization of integers and the Fermat numbers.

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:
4. Charles M Grinsted, J Laurie Snell “Introduction to Probability”, American Mathematical Society
Course Outcomes: After learning the contents of this paper the student must be able to

CO1: Compute probabilities using theorems in probability and probability distributions of single and multiple random variables.
CO2: Apply Inferential Statistics to make predictions or judgments about the population from which the sample data is drawn.
CO3: Establish relationships between variables using correlation and regression
CO4: Apply the number theory concepts to cryptography domain.

<table>
<thead>
<tr>
<th>Course Outcomes (COs)</th>
<th>Program Outcomes (POs)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PO1</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
Data Structures and Algorithms

Prerequisite: Basic mathematical, analytical and logical capability, problem solving through C.

Course Objectives:

- Introduce Analysis of Algorithm in terms of space and time complexity.
- Exploring basic data structures such as stacks and queues.
- Introduces a variety of data structures such as hash tables, search trees, tries, heaps, graphs.
- Introduces sorting and pattern matching algorithms.

Course Outcomes:

- Understand concepts of ADT and to write an algorithm for a given problem statement, also calculate time and space complexity.
- Ability to develop C programs for computing real-life applications using data structures like linked lists, stacks, queues.
- Understand different types of trees like Red-Black, AVL, Splay trees.
- Ability to implement searching and sorting algorithms, and pattern matching algorithms.

Module I

Introduction: Basics of Data Structures, Abstract data types, Dynamic aspects of operations on data, Characteristics of data structures, Creation and manipulation of data structures, Operations on data structures.

Introduction to Algorithms: asymptotic notations divide & conquer recursion, analyzing recursive algorithms.

Programs:

1. Asymptotic notations to calculate the running time complexity of any algorithm. O Notation (Big-O Notation), Ω Notation (Big-Omega Notation), θ Notation (Theta Notation)
2. Calculate complexity analysis of control structures
3. Calculate complexity analysis of any recursive algorithm

Module II

Linked lists: Types of linked lists – singly, doubly and circularly linked lists, operations on linked lists.
Stacks: Implementation of stacks– array and linked list, operations on stacks, Applications of Stacks, Notations – infix, prefix and postfix, Conversion and evaluation of arithmetic expressions using Stacks.

Queues: Implementation of queues– array and linked list, operations on queues, Types of queues – queue, double ended queue and priority queue.

Programs:

1. Write a Program to Implement Stack Operations using Dynamic Memory Allocation.
2. Write a program to convert expressions infix to postfix using stack.
3. Write a program to evaluate arithmetic expressions using stack.
4. Write a program that uses functions to perform the following operations on Singly linked list.
   i) Creation    ii) Insertion    iii) Deletion    iv) Traversal
5. Write a program that uses functions to perform the following operations on Circular linkedlist.
   i) Creation    ii) Insertion    iii) Deletion    iv) Traversal
6. Write a program that implement Queue (its operations) using Arrays.
7. Write a program that implement stack (its operations) using Pointers

Module III

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.

Programs:

1. Write a program to implement dictionary linear list representation and its operations
2. Write a Program to Implement Hash Tables with Quadratic Probing
3. Write a Program to Implement Hash Tables with Linear Probing
4. Write a Program to Implement Hash Tables Chaining with Binary Trees
5. Write a Program to Implement Hash Tables Chaining with Doubly Linked Lists
Module- IV

Graph: Basic terminologies and Representation Traversal algorithms Breadth First Search, Depth First Search, Shortest path: Depth first search in directed and undirected graphs. Union-find data structure and applications. Directed acyclic graphs, topological sort.


Programs:
1. Write a program to implement the Binary Search tree.
   a) Insertion    b) Deletion    c) Traversal    d) Searching element in tree.

2. Write a program to implement the AVL tree.
   a) Insertion    b) Deletion    c) Traversal    d) Searching element in tree.

3. Write a program to implement the Red-Black tree.
   a) Insertion    b) Deletion

4. Write a program to implement the Splay tree.
   a) Insertion    b) Deletion

5. Write a program to implement graph traversal methods
   a) Breadth First Search, b) Depth First Search

Module V

Sorting: objective and Properties of different sorting Algorithms Insertion Sort, Bubble sort, Selection Sort, Merge sort, Quick Sort, Heap sort, Radix sort, Bucket sort. Performance and comparison among all the methods


Programs:
1. Write a program to implement merge sort using divide and conquer technique
2. Write a program to implement bubble sort.
3. Write a program to implement brute-force method of string matching.
4. Write a Program to perform string matching using Naive String Matching
5. Write a program to implement Standard Trie
6. Write a program to implement Compressed Trie
7. Write a program to implement Suffix Trie

TEXT BOOKS:


REFERENCE BOOKS:


WEB RESOURCES:

https://www.javatpoint.com/c-programming-language-tutorial

https://www.tutorialspoint.com/cprogramming/index.htm
Course Outcomes:

<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>3</td>
<td>2</td>
</tr>
<tr>
<td>CO2</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>CO3</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>CO4</td>
<td>3</td>
<td>2</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>
Analog and Digital Electronics

Prerequisite: Basic Electrical and Electronics Engineering

Course Objectives:
1. To familiarize the student with the principal of operation, analysis and design of junction diode and BJT
2. To understand basic number systems codes, logical gates & simplifying Boolean expressions
3. Design and analyze combinational and sequential circuits.
4. Know the characteristics of various electronic components.

Course Outcomes: Upon completing this course, the student will be able to
1. To familiarize the student with the principal of operation, analysis and design of junction diode and BJT
2. To understand basic number systems codes, logical gates & simplifying Boolean expressions
3. Design and analyze combinational and sequential circuits.
4. Know the characteristics of various electronic components

Module-I
P-N Junctions: P-N junction diode working, I-V characteristics of a diode, half-wave and full-wave rectifiers, clamping and clipping circuits, Special purpose diode - Zener diode, Varactor diode.

Bipolar Junction Transistors (BJT): Transistor fundamentals and operations, Input output characteristics of BJT in CB, CE, CC configurations, DC operating point, Load line analysis, common-emitter, common-base and common collector amplifiers; design and analysis of the fixed bias, emitter bias with and without emitter resistance circuits, variation of operating point and its stability.

Experiments:
- PN Junction diode characteristics A) Forward bias B) Reverse bias.
- Design and implement Half Wave Rectifier & Full Wave Rectifier with & without filters
- Design and observe Zener diode characteristics in forward and reverse bias condition. Realize its application as voltage regulator.
- Design and realize input and output characteristics of common emitter amplifier of BJT.
Module – II

Field Effect transistors: JFET- current-voltage characteristics and issues related to its performance, MOSFET structure, I-V characteristics of MOSFET, MOSFET as a switch, small signal equivalent circuits - gain, input and output impedance


Experiments:
● Realize the Input and output characteristics of FET in common source configuration.
● Drain and transfer characteristics of MOSFET
● Frequency response of CE amplifier
● Inverting and non-inverting amplifier

Module - III

Feedback and Oscillator Circuits: Concepts of feedback, Effect of positive and negative feedbacks, basic feedback topologies & their properties, Condition for Oscillations, RC type Oscillators-RC phase shift and Wien-bridge Oscillators, LC type Oscillators –Generalized analysis of LC Oscillators, Hartley and Colpitts Oscillators.

Fundamentals of Digital Systems: Decimal, binary, octal, hexadecimal number systems and their conversion, binary weighted codes, signed numbers, 1s and 2s complement codes, error detecting and correcting codes, Binary arithmetic, Binary logic functions, Boolean laws, truth tables, associative and distributive properties, De Morgan's theorems, digital circuits, AND, OR, NOT, NAND, NOR and Exclusive- OR operations.

Experiments:
● RC Phase Shift and Hartley Oscillator
● Design a bistable multivibrator and draw its waveforms
● Design a astable multivibrator and draw its waveforms
● Wien Bridge Oscillator

Module IV

Minimization of Boolean Expressions: Standard representation for logic functions, SOP, POS, K-map representation, and simplification of logic functions using K-map, Quine-McCluskey minimization technique.

Combinational Circuits: Adders, Subtractors, BCD arithmetic, carry look a head adder, serial adder, digital comparator, priority encoders, decoders, ALU.

Experiments:
● Design and realization of Boolean expression using basic gates & universal gates
● Design and implementation of half adder and full adder using universal gates.
● Design and implement of half sub tractor and full sub tractor using basic gates.
● Design and implement code converter I) Binary to Gray (II) Gray to Binary Code using basic gates.
Module V

Sequential Circuits: Sequential circuits, flip-flops, latches, the clocked SR flip flop, J, K, T and D types flip-flops, State diagrams and tables, transition table, excitation table and equations, applications of flip-flops.

Applications of Sequential circuits: Shift registers, applications of shift registers, serial to parallel converter, parallel to serial converter, ring counter, sequence generator, ripple (Asynchronous) counters, synchronous counters.

Experiments:

- Design a 4 – bit Adder/Subtractor and realization of 4-bit comparator
- Design and realization of a 4 – Bit Gray to Binary and Binary to Gray Converter
- Design and realization of 8x1 MUX using 2x1MUX
- Design and realization of an 8-bit parallel load and serial out shift register using flipflops.
- Design and realization of a Synchronous and Asynchronous counter using flip-flops

Text Books:


Reference Text Books:


Web resources:

1. https://www.youtube.com/watch?v=yQDfVJzEyMI
2. https://www.tutorialspoint.com/
3. https://www.youtube.com/watch?v=2xXErGeeb_Q

<table>
<thead>
<tr>
<th>CO-PO/PSO Mapping Chart</th>
</tr>
</thead>
<tbody>
<tr>
<td>(3/2/1 indicates strength of correlation)</td>
</tr>
<tr>
<td>3 – High; 2 – Medium; 1 - Low</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cours e Outco mes (COs)</th>
<th>Program Outcomes (POs)</th>
<th>Program Specific Outcomes*</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PO1</td>
<td>PO2</td>
</tr>
<tr>
<td>CO1</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>CO2</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>CO3</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>CO4</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>
DOING ENGINEERING-1

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

1. Describe the controllers working.
2. Analysis the cloud based projects
3. Analyse the real time monitoring system.
4. 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:


<table>
<thead>
<tr>
<th>CO-PO/PSO Mapping Chart (3/2/1 indicates strength of correlation)</th>
<th>3 – High; 2 – Medium; 1 - Low</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Course Outcomes (COs)</strong></td>
<td><strong>Program Outcomes (POs)</strong></td>
</tr>
<tr>
<td>CO1</td>
<td>PO 1</td>
</tr>
<tr>
<td>3</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>
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.

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.

Module I
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.

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

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’

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

**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,

**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

**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

**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


**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, andothers… Telugu Academy, Telangana Gov. 2015

**References:**


**MOOC Course:** NPTEL -Exploring Human Values: Visions of Happiness and Perfect
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
B. Tech II Year–II Sem

Subject Code: 22PC4CO02

3 0 0 3

Computer Organization and Architecture

Pre-requisite: Knowledge of Computers

Course Objective:

1. The purpose of the course is to introduce principles of computer organization and the basic architectural concepts and it also introduces simple register transfer language to specify various computer operations.
2. Understand the representation fixed-point and floating-point numbers in computer and develop hardware algorithms using them for fixed-point and floating-point arithmetic.
3. The course would display understanding of instruction set of RISC processor and develop understanding of how memory is organized and managed in a modern digital computer, including cache, virtual and physical memory.
4. It discusses input-output units and how they communicate with the processor, and how their performance is computed.

Course Outcomes: Upon completing this course, the student will be able to

1. Interpret the basics of instructions sets and their impact on processor design.
2. Demonstrate representations of numbers stored and 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.

Module 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.

Module II:

Number representation: Binary Data representation, Signed Number representation, Fixed and Floating-point data representations. IEEE 754 floating point number representation.

Module III:

Control Unit: Hardwired controls, Micro programmed Control: Control memory, Address sequencing, micro program example, design of control unit.
Central Processing Unit: Program control, Reduced instruction set computer, Complex instruction set computer, Data Transfer, Manipulation, Addressing mode, General Register Organization, Instruction Formats.

Module IV:

Input-Output Organization: Input output interface, Data transfer schemes, Direct memory access transfer, Asynchronous data transfer, Modes of Transfer, Priority Interrupt Direct memory Access.
Memory Organization: Memory Hierarchy, Main Memory, Auxiliary memory, Associate Memory, Cache Memory.

Module V:

Pipeline and Vector Processing: Parallel Processing, Pipelining, Arithmetic Pipeline, Instruction Pipeline, RISC Pipeline, Vector Processing, Array Processor.
Multi Processors: Characteristics of Multiprocessors, Interconnection Structures, Inter Processor arbitration, bus arbitration, and synchronization, Cache Coherence.
Introduction to Inter-processor communication(IPC): Basics of Inter-processor communication, Working principle of Inter-processor communication. Different ways of IPC are pipe, message passing, message queue, shared memory, direct communication, indirect communication, and FIFO

Text Books:

References:

CO-PO/PSO Mapping Chart:

<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>CO1</td>
<td>3</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>
<td></td>
</tr>
<tr>
<td>CO2</td>
<td>3</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>
<td></td>
</tr>
<tr>
<td>CO3</td>
<td>3</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>
<td></td>
</tr>
<tr>
<td>CO4</td>
<td>3</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>
<td></td>
</tr>
</tbody>
</table>
Design and Analysis of Algorithms

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.
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 paths to other vertices using Dijkstra's algorithm.
2. Write a C program to implement the Stack using arrays. Write Push(), Pop(), and 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 \( S = \{ S_1, S_2, \ldots, S_n \} \) 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>CO</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>
</tr>
</thead>
<tbody>
<tr>
<td>PO</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
</tr>
<tr>
<td>PSO</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
</tr>
</tbody>
</table>

| CO1 | 3 | 2 | 3 | 3 |
| CO2 | 3 | 2 | 3 | 3 |
| CO3 | 3 | 2 | 3 | 3 |
| CO4 | 3 | 2 | 3 | 3 |
Prerequisite: Python programming, ADE

Course Objectives:

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

Course Outcomes:

Upon completing this course, the student will be able to

1. Understand the IoT value chain structure (device, data cloud), application areas and technologies involved.
2. Analyse IoT sensors and technological challenges faced by IoT devices, with a focus on wireless, energy, power, and sensing modules
3. Analyse usage of sensors and devices in IoT applications
4. Explore Internet of Things with the help of preparing projects designed for Raspberry Pi

MODULE-I:

Program:

- Connect an LED to GPIO pin 25 and control it through the command line.
- Connect an LED to GPIO pin 24 and a Switch to GPIO 25 and control the LED with the switch
- Create an application that has three LEDs (Red, Green and white). The LEDs should follow the cycle (All Off, Red On, Green On, White On) for each clap (use sound sensor)

MODULE-II:

Controlling Hardware- Connecting LED, Buzzer, Switching High Power devices with transistors, speed control of DC Motor, unipolar and bipolar Stepper motors Sensors- Light sensor, temperature sensor with thermistor, voltage sensor, ADC and DAC, Temperature and Humidity Sensor DHT11, Motion Detection Sensors, Bluetooth, Level Sensors, USB.
Program:
- Use joystick and display the direction on the screen
- Use Light Dependent Resistor (LDR) and control an LED that should switch-on/off depending on the light.
- Create a traffic light signal with three coloured lights (Red, Orange and Green) with a duty cycle of 5-2-10 seconds.

MODULE-III:


Program:
- Control a 230V device (Bulb) with Raspberry Pi using a relay.
- Control a 230V device using a threshold temperature, using a temperature sensor.
- Create a door lock application using a reed switch and magnet and give a beep when the door is opened

MODULE-IV:

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

Program:
- Switch on and switch of a DC motor based on the position of a switch.
- Convert an analog voltage to digital value and show it on the screen.
- The state of LED should toggle with every press of the switch Use DHT11 temperature sensor and print the temperature and humidity of the room with an interval of 15 seconds

MODULE-V:

Introduction to clouds– Introduction to Cloud computing, Services and deployment models, Storage models , Communication models and Communication APIs – Web server for IoT, Cloud for IoT

Program:
- IoT based Create a Smart attendance system .
- IoT based System accessing in laboratory.
TEXT BOOKS:


REFERENCE BOOKS:


<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>P O1</td>
<td>P O2</td>
</tr>
<tr>
<td>CO1</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>CO2</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>CO3</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>CO4</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

CO-PO/PSO Mapping Chart
(3/2/1 indicates strength of correlation)
3 – High; 2 – Medium; 1 - Low
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></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>
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
Writing- 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>3</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