# HYDERABAD INSTITUTE OF TECHNOLOGY AND MANAGEMENT
## B.TECH. HR-2021 COURSE STRUCTURE - CSE

(Applicable from the batch admitted during 2021-22 and onwards)

### III – Semester (II – Year)

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Course code</th>
<th>Subject</th>
<th>Hours Per Week</th>
<th>Credits</th>
<th>Scheme of Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>L  T  P</td>
<td></td>
<td>Maximum Marks</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Int.   Ext.  Tot.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>21BS3MT03</td>
<td>Probability and Statistics</td>
<td>3  1  0</td>
<td>4</td>
<td>30 70 100</td>
</tr>
<tr>
<td>2</td>
<td>21ES3CS03</td>
<td>Python Programming</td>
<td>3  0  0</td>
<td>3</td>
<td>30 70 100</td>
</tr>
<tr>
<td>3</td>
<td>21PC3CS01</td>
<td>Data Structures</td>
<td>3  1  0</td>
<td>4</td>
<td>30 70 100</td>
</tr>
<tr>
<td>4</td>
<td>21PC3CS17</td>
<td>Discrete Mathematics</td>
<td>3  0  0</td>
<td>3</td>
<td>30 70 100</td>
</tr>
<tr>
<td>5</td>
<td>21PC3CS02</td>
<td>Database Management Systems</td>
<td>3  0  0</td>
<td>3</td>
<td>30 70 100</td>
</tr>
<tr>
<td>6</td>
<td>21ES3CS04</td>
<td>Python Programming Lab</td>
<td>0  0  3</td>
<td>1.5</td>
<td>30 70 100</td>
</tr>
<tr>
<td>7</td>
<td>21PC3CS05</td>
<td>Data Structures Lab</td>
<td>0  0  2</td>
<td>1</td>
<td>30 70 100</td>
</tr>
<tr>
<td>8</td>
<td>21PC3CS04</td>
<td>Database Management Systems Lab</td>
<td>0  0  3</td>
<td>1.5</td>
<td>30 70 100</td>
</tr>
<tr>
<td>9</td>
<td>21PR3IN01</td>
<td>Internship-1</td>
<td>1</td>
<td>100</td>
<td>- 100</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TOTAL</td>
<td>22</td>
<td>340 560 900</td>
<td></td>
</tr>
</tbody>
</table>

**Mandatory Course (Non-Credit)**

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Course code</th>
<th>Subject</th>
<th>Hours Per Week</th>
<th>Credits</th>
<th>Scheme of Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>L  T  P</td>
<td></td>
<td>Maximum Marks</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Int.   Ext.  Tot.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>21MC3HS01</td>
<td>Universal Human Values</td>
<td>2  0  0</td>
<td>0</td>
<td>100 - 100</td>
</tr>
</tbody>
</table>

### IV – Semester (II – Year)

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Course code</th>
<th>Subject</th>
<th>Hours Per Week</th>
<th>Credits</th>
<th>Scheme of Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>L  T  P</td>
<td></td>
<td>Maximum Marks</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Int.   Ext.  Tot.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>21HS4EG03</td>
<td>English for Employability</td>
<td>1  0  0</td>
<td>1</td>
<td>30 70 100</td>
</tr>
<tr>
<td>2</td>
<td>21PC4CS18</td>
<td>Computer Organization and Architecture</td>
<td>3  0  0</td>
<td>3</td>
<td>30 70 100</td>
</tr>
<tr>
<td>3</td>
<td>21PC4CS07</td>
<td>Design and Analysis of Algorithms</td>
<td>3  0  0</td>
<td>3</td>
<td>30 70 100</td>
</tr>
<tr>
<td>4</td>
<td>21PC4CS14</td>
<td>Formal Languages and Automata Theory</td>
<td>3  0  0</td>
<td>3</td>
<td>30 70 100</td>
</tr>
<tr>
<td>5</td>
<td>21PC4CS03</td>
<td>Object Oriented Programming using Java</td>
<td>3  1  0</td>
<td>4</td>
<td>30 70 100</td>
</tr>
<tr>
<td>6</td>
<td>21PC4CS19</td>
<td>Software Engineering</td>
<td>3  0  0</td>
<td>3</td>
<td>30 70 100</td>
</tr>
<tr>
<td>7</td>
<td>21HS4EG04</td>
<td>English for Employability-Lab</td>
<td>0  0  2</td>
<td>1</td>
<td>30 70 100</td>
</tr>
<tr>
<td>8</td>
<td>21PC4CS06</td>
<td>Programming using Java Lab</td>
<td>0  0  3</td>
<td>1.5</td>
<td>30 70 100</td>
</tr>
<tr>
<td>9</td>
<td>21PC4CS15</td>
<td>Software Engineering Lab</td>
<td>0  0  2</td>
<td>1</td>
<td>30 70 100</td>
</tr>
<tr>
<td>10</td>
<td>21PR4CS01</td>
<td>Doing Engineering-1</td>
<td>1  0  1</td>
<td>1.5</td>
<td>30 70 100</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TOTAL</td>
<td>22</td>
<td>300 700 1000</td>
<td></td>
</tr>
</tbody>
</table>
PROBABILITY AND STATISTICS

Pre-requisite: Basic knowledge of set and relations theory, permutations, combinations, Venn diagrams, mean, median, mode and testing of hypothesis of different samples

Course Objectives: To provide the student with

Different random variables and various probability distribution functions.
- The discrete and continuous of the Bernoulli, binomial, Poisson and the Normal distributions.
- Measures of Central tendency and curve fitting by the method of least squares.
- Concept of point and interval estimations and concept of test of hypothesis and test of significance.
- How to apply the five-step test procedure for test of hypothesis concerning a population mean when the sample size is small.

MODULE I

Unit 1: PROBABILITY

Unit 2: RANDOM VARIABLES

MODULE II

Unit 1: DISCRETE DISTRIBUTIONS
Uniform Distribution, Binomial Distribution, Poisson distribution and statistical constants of these distributions using moment generating function.

Unit 2: CONTINUOUS DISTRIBUTIONS
Uniform Distribution, Exponential Distribution, Normal Distribution, and statistical constants of these distributions using moment generating function.

MODULE III

Unit 1: BIVARIATE DISTRIBUTION
Joint Probability distributions - Joint Probability mass function, joint probability density function, Marginal Distribution, Covariance of two random variables.

Unit 2: CORRELATION and REGRESSION
Karl Pearson coefficient of correlation, Rank correlation, Regression coefficient, Lines of regression.
MODULE IV
Unit 1: SAMPLING DISTRIBUTION:
Random Sampling, Some Important Statistics, Sampling Distributions, Sampling Distribution of Means, variance and the Central Limit Theorem.
Unit 2: ESTIMATION:

MODULE V
Unit 1: TESTS OF HYPOTHESES:
Null & Alternative Hypothesis, Critical region, Type I and Type II errors, level of significance, one tail, two-tail tests. Large sample test for single mean, difference of means, single proportion, difference of proportions.
Unit 2: TEST 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.

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: Solve the probability and the probability distribution function of random variables.
CO2: Evaluate random processes which occur in engineering applications governed by the Binomial, Poisson, Normal and Exponential 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: Calculate Correlation coefficient and Regression coefficients of the given bivariate Data.

<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>
Pre-requisite: Programming for Problem solving Using C

Course Objectives:
1. Acquire programming skills in core Python.
2. Acquire Object-oriented programming skills in Python.
3. Develop the skill of designing graphical-user interfaces (GUI) in Python.
4. Develop the ability to write database applications in Python.

Course Outcomes:
1. Understand operators, precedence of operators, associatively while evaluating expressions in program statements.
2. Distinguish the capabilities of procedural as well as object-oriented programming in Python and demonstrate the same in real world scenario.
3. Demonstrate indexing and slicing mechanisms for extracting a portion of data in a sequence.
4. Extend the knowledge of Python programming to build successful career in software development.

Module I
Python Basics, Objects- Python Objects, Standard Types, Other Built-in Types, Internal Types, Standard Type Operators, Standard Type Built-in Functions, Categorizing the Standard Types, Unsupported Types
Numbers - Introduction to Numbers, Integers, Floating Point Real Numbers, Complex Numbers, Operators, Sequences - Strings, Lists, and Tuples, Mapping and Set Types

Module II
Conditional Statements: Selection/Conditional Branching Statements: if, if-else, nested if, if-elif-else statement(s).
Iterative Statements – while and for loop, Nested loops, break and continue statement, pass Statement, else Statement used with loops.

Module III
Functions: Communicating with functions, Variable Scope and lifetime, return statement, Types of arguments, Lambda functions, and Recursive functions
Classes and Objects – Defining Classes, Creating Objects, Data Abstraction and Hiding through Classes, Class Method and self-Argument, Class variables and Object variables, __init()__ and __del__() method.

Module IV
Exceptions: Exceptions in Python, Detecting and Handling Exceptions, Context Management, , Raising Exceptions, Related Modules
Regular Expressions: Introduction, Special Symbols and Characters, Res and Python
Module V
GUI Programming: Introduction, Tkinter and Python Programming, Brief Tour of Other GUIs, Related Modules and Other GUIs
Text Books:

Reference Books:

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>M</td>
<td></td>
</tr>
<tr>
<td>CO2</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>CO3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

CO-PO/PSO Mapping Chart
(3/2/1 indicates strength of correlation)
3 – High; 2 – Medium; 1 - Low
Subject Code: 21PC3CS01

DATASTRUCTURES

Prerequisite: Knowledge of C Programming

Course Objectives:
1: To make student knowledgeable in data structure concepts
2: To make students apply the data structure concepts in relevant applications
3: To make students apply the graph techniques in relevant applications
4: To make students determine the best sorting technique to process the data

Course Outcomes:
At the end of the course the student s will be able to:
1: Identify the concepts of data structures in relevant applications
2: Apply the concept of data structures for the new situations
3: Apply the graph traversal techniques to solve map related problems
4: Analyze the best sorting technique pertaining to the area applied

Module 1:
Algorithm Analysis: Asymptotic analysis, simple justification techniques, Recursion, Illustrative examples, Analyzing Recursive algorithms, Designing Recursive algorithms.
Sorting: Bubble sort, selection sort, Insertion sort, Merge sort, Quick Sort, Randomized Quick sort. Comparison of sorting algorithms, Polynomials and Sparse Matrices.

Module 2:

Module 3:
Linked Lists – Singly Linked List, Implementation using arrays, Implementation using Queues, Circular Linked List- Doubly linked list- Implementation of doubly linked list, implementation using De queue.
Trees- General trees, Binary trees, Implementation using array and linked representation, tree traversals- in order, preorder, post order, Breadth-first traversal, Implementation, Application of tree traversals.

Module 4:
Priority Queue-ADT, Implementation, Heap, sorting with a priority Queue, Adaptable priority Queues Hash Tables and Skip Lists-Maps and Dictionaries, Hash tables, Sorted Maps, Skip lists, Sets, Multi sets and Multi maps
Graphs: Definitions, Terminologies, Matrix and Adjacency List Representation of Graphs, Traversal methods: Breadth First Search and Depth First Search.
Module 5:

Text Books:

Reference Books:
2. Data Structures using C by E. Balaguruswamy.
3. Data Structures, S. Lipscutz Schumaum; Outlines, TMH

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>H</td>
<td>M</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>M</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO-2</td>
<td>M</td>
<td>M</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>L</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO-3</td>
<td>M</td>
<td>M</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>M</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO-4</td>
<td>M</td>
<td>M</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>M</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Discrete Mathematics

Prerequisite: Intermediate Mathematics

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

Course Outcomes:
1. Evaluate the notions of propositions, predicate formulae, Rules of inference.
2. Describe various types of relations, Sets, functions and Classify a mathematical problem into its algebraic structure.
3. Ability to solve counting problems and manipulate sequences and recurrence relations.
4. Ability to use tree and graph algorithms to solve problems.

Module I
Predicates: Predicates, Quantifiers, Nested Quantifiers, Rules of inference, Consistency, proof of contradiction.

Module II
Algebraic Structures: Algebraic systems Examples and general properties, Semi groups and monads, groups sub groups’ homomorphism, Isomorphism.

Module III
Binomial Coefficients: The Binomial and Multinomial Theorems, the Principle of Inclusion-Exclusion.
Module IV

**Counting Techniques:** Recurrence Relations, Solving Linear Recurrence Relations, solving recurrence relation by substitution and Characteristics roots, solution of Inhomogeneous Recurrence Relation.

**Generating functions:** Function of Sequences Calculating Coefficient of generating function, Inclusion-Exclusion, Applications of Inclusion-Exclusion.

Module V

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

**Trees:** Introduction to Trees, Applications of Trees, Tree Traversal, Topological Sorting, Spanning Trees, Minimum Spanning Trees.

**Text Books:**

1. Discrete Mathematical Structures with applications to computer science Trembly J.P. & Manohar, TMH

**Reference Books:**


**Web Resources:**

1. NPTEL
3. [https://www.edx.org/learn/discrete-mathematics](https://www.edx.org/learn/discrete-mathematics)
## 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>PO1</td>
<td>PO2</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>2</td>
<td>3</td>
</tr>
<tr>
<td>CO4</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>
Prerequisite: Basics of computer programming language.

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

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

Module I
Database System Applications: A Historical Perspective, File Systems versus a DBMS, the Data Model, Levels of Abstraction in a DBMS, Data Independence, Structure of a DBMS
Introduction to Database Design: Database Design and ER Diagrams, Entities, Attributes, and Entity Sets, Relationships and Relationship Sets, Additional Features of the ER Model, Conceptual Design with the ER Model

Module II
Relational Model: Integrity constraint over relations, enforcing integrity constraints, querying relational data, logical data base design, introduction to views, destroying/altering tables and views.
Relational Algebra and Calculus: Selection and Projection, Set operations, Joins, Division, More examples on Algebra queries, Tuple relational Calculus, Domain Relational Calculus. Form of Basic SQL Query, Data Definition commands, Data manipulation commands.

Module III
Advanced SQL: SET operators, Data Time Functions, Aggregate Operators, Group by clause, Joins, Advanced Sub queries, Nested Queries, triggers and active data bases.

Module IV
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, Comparison of File Organizations, Indexes and Performance Tuning
Tree Structured Indexing: Intuitions for tree Indexes, Indexed Sequential Access Methods
Module V

**Transaction Management:** The ACID Properties, Transactions and Schedules, Concurrent Execution of Transactions, Lock-Based Concurrency Control, Performance of Locking, Transaction Support in SQL, Introduction to Crash Recovery

**Concurrency Control:** 2PL, Serializability, and Recoverability, Introduction to Lock Management, Lock Conversions, Dealing with Deadlocks, Specialized Locking Techniques, Concurrency Control without Locking

**Text Books:**


**Reference Books:**


**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>3</td>
<td></td>
</tr>
<tr>
<td>CO2</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>CO3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>CO4</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>
Python Programming Lab
(Common To All Branches)

Prerequisite: Basic Programming Skills.

Course Objectives:
To learn the fundamentals of computers.

1. Acquire programming skills in core Python.
2. Acquire Object-oriented programming skills in Python.
3. Develop the skill of designing graphical-user interfaces (GUI) in Python.
4. Develop the ability to write database applications in Python.

Course Outcomes:
1. Acquire programming skills in core Python.
2. Acquire Object-oriented programming skills in Python.
3. Develop the skill of designing graphical-user interfaces (GUI) in Python.
4. Develop the ability to write database applications in Python.

List of Experiments:

Experiment – 1: OPERATORS
a. Read a list of numbers and write a program to check whether a particular element is present or not using membership operators.
b. Read your name and age and write a program to display the year in which you will turn 100 years old.

Experiment – 2: CONTROL STRUCTURES
a. Read your email id and write a program to display the no of vowels, consonants, digits and white spaces in it using if…elif…else statement.
b. Write a Program to find the sum of a Series 1/1! + 2/2! + 3/3! + 4/4! +…….+ n/n!.
   (Input : n = 5, Output : 2.70833)

Experiment – 3: LIST
a. Read a list of numbers and print the numbers divisible by x but not by y (Assume x = 4 and y = 5).
b. Read a list of numbers and print the sum of odd integers and even integers from the list.(Ex: [23, 10, 15, 14, 63], odd numbers sum = 101, even numbers sum = 24)

Experiment – 4: TUPLE
a. Given a list of tuples. Write a program to find tuples which have all elements divisible by K from a list of tuples. test_list = [(6, 24, 12), (60, 12, 6), (12, 18, 21)], K = 6, Output : [(6, 24, 12), (60, 12, 6)]
Experiment – 5: SET
a. Write a program to generate and print a dictionary that contains a number (between 1 and n) in the form (x, x^2).
b. Write a program to perform union, intersection and difference using Set A and Set B.

Experiment – 6: DICIONARY
a. Write a program to do the following operations:
i. Create a empty dictionary with dict( ) method
ii. Add elements one at a time
iii. Update existing key’s value
iv. Access an element using a key and also get( ) method
v. Deleting a key value using del( ) method
b. Write a program to create a dictionary and apply the following methods:
i. pop( ) method
ii. pop item( ) method
iii. clear( ) method

Experiment – 7: STRINGS
a. Given a string, write a program to check if the string is symmetrical and palindrome or not. A string is said to be symmetrical if both the halves of the string are the same and a string is said to be a palindrome string if one half of the string is the reverse of the other half or if a string appears same when read forward or backward.
b. Write a program to read a string and count the number of vowel letters and print all letters except 'e' and 's'.

Experiment – 8: USER DEFINED FUNCTIONS
a. Write a function merge_dict(dict1, dict2) to merge two Python dictionaries.
b. Given a list of n elements, write a linear_search() function to search a given element x in a list.

Experiment – 9: BUILT-IN FUNCTIONS
a. Write a program to demonstrate the working of built-in statistical functions mean ( ), mode ( ), median () by importing statistics library.
b. Write a program to demonstrate the working of built-in trigonometric functions sin ( ), cos ( ),
tan ( ), hypot ( ), degrees ( ), radians ( ) by importing math module.

Experiment – 10: CLASS AND OBJECTS
a. Write a program to create a Bank Account class. Your class should support the following methods for
i) Deposit
ii) Withdraw
iii) Get Balance
iv) Pin Change
b. Write a program to create an employee class and store the employee name, id, age, and salary using the constructor. Display the employee details by invoking employee_info () method and also using dictionary (__dict__).
Reference Books:


CO-PO/PSO Mapping Chart:

<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 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>1</td>
<td></td>
</tr>
<tr>
<td>CO2</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>CO3</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>CO4</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>
DATA STRUCTURES LAB

Pre-requisite: Programming in C and Problem Solving

Course Objectives:

1. Understand various data representation techniques in the real world.
2. Implement linear and non-linear data structures.
3. Analyze various algorithms based on their time and space complexity.
4. Develop real-time applications using suitable data structure.
5. Identify suitable data structure to solve various computing problems.

Course Outcomes:

1. Identify the appropriate data structures and algorithms for solving real world problems
2. Implement various kinds of searching and sorting techniques.
3. Implement data structures such as stacks, queues, search trees, and hash tables to solve various computing problems
4. Choose appropriate traversal method to find shortest distance

List of Experiments:

1. Develop a program to read a linear list of items and store it in an array.
   - Copy the contents from one array to another array
   - Copy the contents from one array to another array in reverse order
   - Delete the duplicate elements from an array.

2. Develop a program to implement various sorting techniques:
   - Insertion sort
   - Selection Sort
   - Bubble Sort
   - Merge Sort
   - Quick Sort

3. Develop a program to create a stack and perform various operations on it.
   - Develop a program to create a queue and perform various operations on it.

4. Develop programs for the following:
   - Uses Stack operations to convert infix expression into postfix expression.
   - Uses Stack operations for evaluating the postfix expression.

5. Design a program to create a singly linked list for the following operations
   - Insert a Node at Beginning, at Ending and at a given Position
   - Delete a Node at Beginning, at Ending and at a given Position
   - Search, Count the Number of Nodes and Display
6. Design a program to create a doubly linked list for the following operations
   a. Insert a Node at Beginning, at Ending and at a given Position
   b. Delete a Node at Beginning, at Ending and at a given Position
   c. Search, Count the Number of Nodes and Display

7. Design a program to create a binary tree and perform various traversals.
   a. In order
   b. Pre Order
   c. Post Order

8. Develop a program to Perform Linear Search and Binary Search on a list stored in an array.

9. Design a program to create a Hash table for the following operations
   a. Insert
   b. Delete
   c. Search

10. Develop programs to implement the following graph traversal algorithms:
    a. Depth first search.
    b. Breadth first search.

Text/Reference Books:
2. Data structures using C - A.S. Tanenbaum, Y. Langsam, and MJ Augestien, PHI Pearson Education

CO-PO/PSO Mapping Chart:

<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 O 1</td>
<td>PS O 1</td>
</tr>
<tr>
<td></td>
<td>P O 2</td>
<td>PS O 2</td>
</tr>
<tr>
<td></td>
<td>P O 3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>P O 4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>P O 5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>P O 6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>P O 7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>P O 8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>P O 9</td>
<td></td>
</tr>
<tr>
<td></td>
<td>P O 10</td>
<td></td>
</tr>
<tr>
<td></td>
<td>P O 11</td>
<td></td>
</tr>
<tr>
<td></td>
<td>P O 12</td>
<td></td>
</tr>
<tr>
<td>CO1</td>
<td>H</td>
<td>M</td>
</tr>
<tr>
<td>CO2</td>
<td></td>
<td>M</td>
</tr>
<tr>
<td>CO3</td>
<td></td>
<td>M</td>
</tr>
<tr>
<td>CO4</td>
<td>H</td>
<td></td>
</tr>
</tbody>
</table>
Subject Code: 21PC3CS04

Database Management System lab

Prerequisite: Nil

Course Objectives:

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

Course Outcomes:

1. Design database schema for a given application and apply normalization.
2. Perform operations on the database using SQL commands for data definition and data manipulation.
3. Develop solutions for database applications using procedures, cursors and triggers.
4. Implement joins on the tables.

List of Experiments:

The following programs are to be developed using My SQL / SQL / PL/SQL

Roadway Travels: "Roadway Travels" is in business since 1997 with several buses connecting different places in India. Its main office is located in Hyderabad.

The company wants to computerize its operations in the following areas:
- Reservations and Ticketing
- Cancellations

Reservations & Cancellation: Reservations are directly handled by booking office. Reservations can be made 30 days in advance and tickets issued to passenger. One Passenger/person can book many tickets (to his/her family).
Cancellations are also directly handed at the booking office.

In the process of computerization of Roadway Travels you have to design and develop a Database which consists the data of Buses, Passengers, Tickets, and Reservation and cancellation details. You should also develop query’s using SQL to retrieve the data from the database.

The above process involves many steps like 1. Analyzing the problem and identifying the Entities and Relationships, 2. E-R Model 3. Relational Model 4. Normalization 5. Creating the database 6. Querying. Students are supposed to work on these steps week wise and finally create a complete “Database System” to Roadway Travels. Examples are given at every experiment for guidance to students.
Experiment 1: E-R Model
Analyze the 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.
Example:
**Entities:** 1. BUS 2. Ticket 3. Passenger  
**Relationships:** 1. Reservation 2. Cancellation  
**Primary Key Attributes:** 1. Ticket ID (Ticket Entity)  
2. Passport ID (Passenger Entity)  
3. Bus_NO (Bus Entity)  
Apart from the above mentioned entities you can identify more. The above mentioned are few.  
**Note:** The student is required to submit a document by writing the Entities and Keys to the lab teacher.

Experiment 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.

Example: E-R diagram for bus

![E-R Diagram for Bus](image)

**Note:** The student is required to submit a document by drawing the E-R Diagram to the lab teacher.
**Experiment 3: Relational Model**
Represent all the entities (Strong, Weak) in tabular fashion. Represent relationships in a tabular fashion. There are different ways of representing relationships as tables based on the cardinality. Represent attributes as columns in tables or as tables based on the requirement.

Different types of attributes (Composite, Multi-valued, and Derived) have different way of representation.

**Example:** The passenger tables look as below. This is an example. You can add more attributes based on your E-R model. This is not a normalized table.

<table>
<thead>
<tr>
<th>Name</th>
<th>Age</th>
<th>Sex</th>
<th>Address</th>
<th>Ticket_id</th>
<th>Passport ID</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Note: The student is required to submit a document by Represent relationships in a tabular fashion to the lab teacher.*

**Experiment 4: Normalization**
Database normalization is a technique for designing relational database tables to minimize duplication of information and, in so doing, to safeguard the database against certain types of logical or structural problems, namely data anomalies. For example, when multiple instances of a given piece of information occur in a table, the possibility exists that these instances will not be kept consistent when the data within the table is updated, leading to a loss of data integrity. A table that is sufficiently normalized is less vulnerable to problems of this kind, because its structure reflects the basic assumptions for when multiple instances of the same information should be represented by a single instance only.

For the above table in the First normalization we can remove the multi valued attribute Ticket_id and place it in another table along with the primary key of passenger.

**First Normal Form: The above table can be divided into two tables as shown below.**

<table>
<thead>
<tr>
<th>Name</th>
<th>Age</th>
<th>Sex</th>
<th>Address</th>
<th>Passport ID</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Passport ID</th>
<th>Ticket_id</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

You can do the second and third normal forms if required. Anyhow Normalized tables are given at the end.
Experiment 5: Installation of Mysql / SQL and practicing DDL commands
Installation of Mysql / SQL. In this session you will learn Creating databases, How to create tables, altering the database, dropping tables and databases if not required. You will also try truncate, rename commands etc.

Example for creation of a normalized “Passenger” table.
CREATE TABLE Passenger (  
    Passport_id INTEGER PRIMARY KEY,  
    Name VARCHAR (50) Not NULL,  
    Age INTEGER Not NULL,  
    Sex Char,  
    Address VARCHAR (50) Not NULL);  

Similarly create all other tables.

Note: Detailed creation of tables is given at the end.

Experiment 6: Practicing DML commands
DML commands are used to for managing data within schema objects. Some examples:
- SELECT - retrieve data from the a database
- INSERT - insert data into a table
- UPDATE - updates existing data within a table
- DELETE - deletes all records from a table, the space for the records remain

Inserting values into “Bus” table:
Insert into Bus values (1234,'hyderabad', 'tirupathi');  
Insert into Bus values (2345,'hyderabd', 'Banglore');  
Insert into Bus values (23,'hyderabd','Kolkata');  
Insert into Bus values (45,'Tirupathi','Banglore');  
Insert into Bus values (34,'hyderabd','Chennai');

Inserting values into “Passenger” table:
Insert into Passenger values (1, 45,'ramesh', 45,'M', 'abc123');  
Insert into Passenger values (2, 78,'geetha', 36,'F', 'abc124');  
Insert into Passenger values (45, 90,'ram', 30,'M', 'abc12');  
Insert into Passenger values (67, 89,'ravi', 50,'M', 'abc14');  
Insert into Passenger values (56, 22,'seetha', 32,'F', 'abc55');

Few more Examples of DML commands:
Select * from Bus; (selects all the attributes and display)  
UPDATE BUS SET Bus No = 1 WHERE BUS NO=2;
Experiment 7: Querying
In this week you are going to practice queries (along with sub queries) using ANY, ALL, IN, Exists, NOT EXISTS, UNION, INTERSECT, Constraints etc.

Practice the following Queries:
1. Display unique PNR_no of all passengers.
2. Display all the names of male passengers.
3. Display the ticket numbers and names of all the passengers.
4. Find the ticket numbers of the passengers whose name start with ‘r’ and ends with ‘h’.
5. Find the names of passengers whose age is between 30 and 45.
6. Display all the passengers names beginning with ‘A’
7. Display the sorted list of passengers names

Experiment 8 and Experiment 9: Querying (continued…)
You are going to practice queries using Aggregate functions (COUNT, SUM, AVG, and MAX and MIN), GROUP BY, HAVING and Creation and dropping of Views.
1. Write a Query to display the Information present in the Passenger and cancellation tables. **Hint:** Use UNION Operator.
2. Display the number of days in a week on which the 9W01 bus is available.
3. Find number of tickets booked for each PNR_no using GROUP BY CLAUSE. **Hint:** Use GROUP BY on PNR_No.
4. Find the distinct PNR numbers that are present.
5. Find the number of tickets booked by a passenger where the number of seats is greater than 1. **Hint:** Use GROUP BY, WHERE and HAVING CLAUSES.
6. Find the total number of cancelled seats.

Experiment 10: Triggers
In this session you are going to work on Triggers. Creation of insert trigger, delete trigger, update trigger. Practice triggers using the above database.

**Eg:**
```
CREATE OR REPLACE TRIGGER SUN_TRI
AFTER INSERT OR UPDATE OR DELETE ON EMP
DECLARE
DY VARCHAR2 (200);
BEGIN
  DY:=TO_CHAR(SYSDATE,'DY');
  IF DY='SUN' THEN
    RAISE_APPLICATION_ERROR (-20005,'TODAY IS SUNDAY TRANSACTION NOT ALLOWED TODAY');
  END IF;
END;
```

Experiment 11: Procedures
In this session you are going to learn Creation of stored procedure, Execution of procedure and modification of procedure. Practice procedures using the above database.

**Eg:**
```
CREATE PROCEDURE myProc()BEGIN
  SELECT COUNT (Tickets) FROM Ticket WHERE age>=40;END;
```
Experiment 12: Cursors
In this session you need to do the following: Declare a cursor that defines a result set.
Open the cursor to establish the result set. Fetch the data into local variables as needed from the
cursor, one row at a time. Close the cursor when done

Eg:
DECLARE
CURSOR A IS
SELECT * FROM EMP;
B A%ROWTYPE;
BEGIN
OPEN A;
LOOP
FETCH A INTO B;
EXIT WHEN A%NOTFOUND;
DBMS_OUTPUT.PUT_LINE('EMP NO IS ' || B.EMPNO);
DBMS_OUTPUT.PUT_LINE('ENAME IS ' || B.ENAME);
DBMS_OUTPUT.PUT_LINE('SAL IS ' || B.SAL);
DBMS_OUTPUT.PUT_LINE('MGR NO IS ' || B.MGR);
DBMS_OUTPUT.PUT_LINE('COMM IS ' || B.COMM);
DBMS_OUTPUT.PUT_LINE('HIREDATE IS ' || B.HIREDATE);
DBMS_OUTPUT.PUT_LINE('DEPTNO IS ' || B.DEPTNO);
DBMS_OUTPUT.PUT_LINE('EMP JOB IS ' || B.JOB);
END LOOP;
CLOSE A;
END;

Text Books:
   3rd Edition

Reference Books:
1. Database Systems design, Implementation, and Management, Peter Rob & Carlos Coronel
   7th Edition.
2. Fundamentals of Database Systems, Elmasri Navrate, Pearson Education
3. Introduction to Database Systems, C. J. Date, Pearson Education

CO-PO/PSO Mapping Chart

<table>
<thead>
<tr>
<th>Course Outcomes (COs)</th>
<th>Program Outcomes (POs)</th>
<th>Program 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></td>
</tr>
<tr>
<td>CO2</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>CO3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>CO4</td>
<td>3</td>
<td></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.

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>O-PO Mapping Chart</th>
</tr>
</thead>
<tbody>
<tr>
<td>(3/2/1 indicates strength of correlation) 3 – High; 2 – Medium; 1 – Low</td>
</tr>
<tr>
<td><strong>Course Outcomes (COs)</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>CO1</td>
</tr>
<tr>
<td>CO2</td>
</tr>
<tr>
<td>CO3</td>
</tr>
<tr>
<td>CO4</td>
</tr>
<tr>
<td>CO5</td>
</tr>
</tbody>
</table>
Course Objective:
1. To introduce growing need of English Language Skills for Employability (ELSE).
2. To help the students understand the importance of inter-personal communication, team dynamics and role behavior at the work place
3. To enhance their Behavioural skill and Entrepreneurship skills.

Module I- Wings of Fire
Unit 1: “Orientation” an excerpt from Wings of fire- An Autobiography of Abdul Kalam by Arun Tiwari.
Unit 2: Grammar
Vocabulary- Omission of Articles, Collective Nouns, Prepositions, Collocations.
Reading- Observation Passage, Survey Passage
Writing- Comprehension Passages based on graphs, charts & other illustrations.

Module II 5 Points Someone
Unit 1: The Gift an excerpt from 5 Points Someone by Chetan Bhagat
Unit 2: 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
Unit 1: “Wise Leaders Wanted & Shift Your Perspective: Connect to Your Noble Purpose” an excerpt from From Smart to Wise: Acting and Leading with Wisdom Kaipa, Prasad, and Navi Radjou.
Unit 2: Grammar
Vocabulary- Technical vocabulary, Auxiliaries and Modals,
Reading- Technical Comprehension,
Writing- Creative Resume.

Module IV- Variation Under Nature
Unit 1: “Variation Under Nature” an excerpt from Origin of Species by Charles Darwin
Unit 2: Grammar
Vocabulary- Coherence-Cohesive devices, Figures of speech
Reading- Inferring Reading, Survey, Question, Read, Recite and Review (SQ3R)

Module V- Let's Build a Company: A Start-up Story Minus the Bullshit
Unit 1: Let's Build a Company: A Start-up Story Minus the Bullshit by Harpreet Grover and Vibhore Goyal
Unit 2: 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, Reported Speech: Direct & Indirect Speech, Email Writing, Formal letter writing (Enquiry, Apology, Leave, Request) Notice Writing, Information Transfer, Technical report writing

**Textbooks**

**References**
5. Murphy, R. *Intermediate English Grammar*.

**Course Outcomes:** After undergoing this course, the student will be able to;

- **CO1:** Understand the concept and process of employability skills.
- **CO2:** Demonstrate the employability skills in both verbal and non-verbal communications.
- **CO3:** Apply skill identification strategies to bring out the results on social and industry demands.
- **CO4:** Recognize right professional, Entrepreneurship skills and social ethical values.

<table>
<thead>
<tr>
<th>CO-PO Mapping Chart</th>
<th>(3/2/1 indicates strength of correlation) 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>CO2</td>
<td>2</td>
</tr>
<tr>
<td>CO3</td>
<td></td>
</tr>
<tr>
<td>CO4</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CO1</th>
<th>CO2</th>
<th>CO3</th>
<th>CO4</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>
Computer Organization and Architecture

Pre-requisite: NIL

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 organised 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:
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, General register and stack organization, Addressing mode.
Module IV:
**Input-Output Organization:** Priority interrupt, Peripheral devices, Input output interface, Data transfer schemes, Program control and interrupts, Direct memory access transfer, Input/output processor.

**Memory Unit:** High speed memories, Memory hierarchy, Processor Vs Memory speed, Cache memory, Associative memory, Interleave, Virtual memory, Memory management.

Module V:
**Reduced Instruction Set Computer:** CISC Characteristics, RISC Characteristics.

**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, Inter-processor communication and synchronization, Cache Coherence.

**Text Books:**

**References:**

**CO-PO/PSO Mapping Chart:**

<table>
<thead>
<tr>
<th>Course Outcomes (COs)</th>
<th>Program Outcomes (POs)</th>
<th>Program Specific Outcomes*</th>
</tr>
</thead>
<tbody>
<tr>
<td>CO-PO</td>
<td>1 2 3 4 5 6 7 8 9 10 11 12</td>
<td>PSO1 PSO2 PSO3</td>
</tr>
<tr>
<td>CO-1</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>CO-2</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>CO-3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>CO-4</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>
DESIGN AND ANALYSIS OF ALGORITHMS

Prerequisites: Data Structure, Discrete Mathematics

Course Objectives

- To analyze the performance of algorithms.
- To choose the appropriate data structure & algorithm design method for specific application
- To understand how the choice of data structure & algorithm design method impact the performance of program
- 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

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 FirstSearch , AND/OR Graphs ,game tree, connected components and biconnected components.

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,


Module IV

Dynamic Programming: General method, applications- 0/1 knapsack problem,

All pairs shortest path problem: Travelling sales person problem, Reliability design.
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.

TEXT BOOKS

REFERENCE BOOKS

CO-PO/PSO Mapping:

<table>
<thead>
<tr>
<th>CO-PO/PSO Mapping Chart (3/2/1 indicates strength of correlation)</th>
<th>Program Outcomes Specific Outcomes*</th>
</tr>
</thead>
<tbody>
<tr>
<td>CO1</td>
<td>CO2</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>

Program Outcomes (POs) | Program Specific Outcomes* |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>PO1</td>
<td>PO2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>
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: Alphabet, languages and grammars, Chomsky hierarchy of languages.
Regular languages and finite automata: Deterministic Finite Automata (DFA), nondeterministic finite automata (NFA)
Equivalence of NFA: Equivalence with DFA, NFA with ε - moves, Conversion to NFA without ε - moves, minimization of finite automata, equivalence between FAs, Finite Automata with Outputs – Mealy machine, Moore machine and equivalence.

Module II
Regular Languages and Finite Automata: Regular sets, Regular expressions and languages, Operations on Languages - Union, Concatenation, Kleen Closure, equivalence between finite automata and regular expressions,
Regular grammars: Definition, productions, derivation, right linear and left linear grammars, and equivalence with Regular grammars and finite automata, properties of regular languages, pumping lemma for regular languages

Module III
Context-free languages: Context-Free Grammars (CFG) and Languages (CFL), parse trees, sentential forms, right most and left most derivations of strings, ambiguity in CFG, Language of the grammar
Simplification of Grammar: Simplification of the grammar Left recursion and left factoring in context free grammars, Chomsky and Greibach normal forms, Pumping Lemma for context-free languages, closure properties of CFLs, Decision properties of CFL

Module IV
Pushdown Automata: definition, model, acceptance of CFL, Pushdown Automata (PDA), Acceptance by final state and acceptance by empty stack and its equivalence, Equivalence of CFG and PDA (proofs not required), Nondeterministic Pushdown Automata (NPDA).
Context Sensitive Grammars: Context-Sensitive Grammars (CSG) and languages, Linear Bounded Automata (LBA) and equivalence with CSG
Module V:

**Turing Machines**: Introduction to Turing Machine, Formal Description, Instantaneous description, the language of a Turing machine. Types of Turing machine and halting

**Computability Theory**: Undecidability: Church-Turing Thesis, universal Turing Machine, undecidable problems about languages. LR (0) grammar, decidability of problems, Post’s Correspondence Problem - The classes P and NP.

**Text Books:**

**Reference Books:**

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

**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>3</td>
<td>2</td>
</tr>
<tr>
<td>CO2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>CO3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>CO4</td>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>
Object Oriented Programming through Java
(Common to CSE/CSM/CDS/CSC/CSO/EEE)

Prerequisites: Knowledge of Programming

Course Objectives:
1. Understand fundamental concepts and constructs of Java
2. Implement Different object-oriented Concepts in Java.
3. Write error free program using exception handling and multithreading
4. Develop GUI application

Course Outcomes:
1. Solve the given problem using OOPS technique.
2. Explain the concept of Package and Interfaces.
3. Implement Multi threading and Exception handling in java
4. Develop GUI based application 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, scope and life time of variables, arrays, operators, expressions, control statements, type conversion and casting, simple java programs, concepts of classes, objects, , methods, constructor, this and static keywords , garbage collection, overloading methods, parameter passing, access control, command line arguments, auto boxing, exploring String and String Buffer class
Inheritance: Types of inheritance, member access rules, super and final keywords, the object class and its methods, method overriding, dynamic method dispatch, abstract classes and methods.

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, benefits of exception handling, exception hierarchy, usage of try, catch, throw, throws and finally, built in exceptions, creating own exceptions.

Module III
Multithreading: Differences between multi-threading and multi programming, thread life cycle, creating threads, thread priorities, synchronizing threads, inter thread communication.
I/O Streams: Types of Streams: Byte and character streams, File class, reading and writing files, reading and writing from console, Random access file, automatically close a file, serialization.

Module IV
Applets: Concepts of Applets, differences between applets and applications, life cycle of an applet, 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.

**Module V**

**Swings**: Introduction, limitations of AWT, MVC architecture, components, containers, exploring swing- J Applet, J Frame and J Component, Image Icon, J Label, J Text field, J Button, J Checkbox, J List, J Radio button, J Combo Box, J Tabbed Pane, J Scroll Pane.

**The Collections Framework (java.util)**- Collections overview, Collection Interfaces, Generics The Collection classes- Array List, Linked List, Hash Set, Tree Set, Priority Queue, Array Deque. Accessing a Collection via an iterator, Using an Iterator, The For-Each alternative, Map Interfaces and Classes, Comparators, Collection algorithms, Arrays, The Legacy Classes and Interfaces- Dictionary, Hash table, Properties, Stack, Vector More Utility classes, String Tokenizer, Date, Calendar, Random, Scanner

**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

**Web Resources:**
1. [https://www.w3schools.com/java/](https://www.w3schools.com/java/)
2. [https://www.tutorialspoint.com/java/index.htm](https://www.tutorialspoint.com/java/index.htm)
3. [https://www.javatpoint.com/java-tutorial](https://www.javatpoint.com/java-tutorial)

**E-Books:**
1. java 2 complete reference 9th ed Herbert Schildt.pdf

**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>CO1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>CO2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>CO3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>CO4</td>
<td>3 1 1 1 3</td>
<td>3</td>
</tr>
</tbody>
</table>
Software Engineering

Pre requisite: NIL

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

Course Outcomes:
1. Understand the basic software engineering methods and practices, and its applications.
2. Ability to translate end-user requirements into system and software requirements and structure the requirements in a software requirements document (srd). Implement software engineering layered technology and process-models.
3. Identify appropriate software architectures and patterns to carry out high level design of a system and be able to critically compare alternative choices.
4. Implement project management including planning, scheduling, risk management, testing strategies for quality assurance.

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

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

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

Module IV

Testing Strategies: A strategic approach to software testing, test strategies for conventional software, black-box and white-box testing, validation testing, system testing, the art of debugging.

Product metrics: Software quality, metrics for analysis model, metrics for design model, metrics for source code, metrics for testing, metrics for maintenance.

Module V


Quality Management: Quality concepts, software quality assurance, software reviews, formal technical reviews, statistical software quality assurance, software reliability, the ISO 9000 quality standards

Text Books:
3. The unified modeling language user guide Grady Booch, James Rambaugh, Ivar Jacobson, Pearson Education.

Reference Books:
3. Fundamentals of object-oriented design using UML Meiler page-Jones; Pearson Education

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>3</td>
<td></td>
</tr>
<tr>
<td>CO2</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>CO3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
English For Employability Lab
(Common to CSE/EEE/ECE/CSM/CSD/CSC/CSO/MECH)

Prerequisites:
1. Basic knowledge of interpersonal and communication skills.
2. Enriching the speaking and writing ability
3. Basic grammar rules of (LSRW)
4. Basic Spoken English skills

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.

Activity 1- Narration (Historical places, events, Picture narration, Memorable incidents of life)
Self-Introduction, 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 (Job Consultancy/Role Play)
**Targeted Skills**- Listening-Writing- Speaking

**Course Outcomes:** After undergoing this course, the student will be able to;

**CO1:** **Understand** the explicit and implicit of importance of employability skills.

**CO2:** **Demonstrate** life skills like team work, learning skills, problem solving, attitude, adaptability and flexibility.

**CO3:** **Apply** critical and analytical skills to bring out the solution on problem/case study.

**CO4:** **Recognize** the need of appropriate words, Phrases & functional grammar and apply them in both spoken and written communication.

<table>
<thead>
<tr>
<th>CO-PO Mapping Chart</th>
<th>(3/2/1 indicates strength of correlation) 3 – High; 2 – Medium; 1 – Low</th>
</tr>
</thead>
<tbody>
<tr>
<td>Course Outcomes (COs)</td>
<td>Program Outcomes (POs)</td>
</tr>
<tr>
<td>CO1</td>
<td>PO 1</td>
</tr>
<tr>
<td>CO2</td>
<td>2</td>
</tr>
<tr>
<td>CO3</td>
<td></td>
</tr>
<tr>
<td>CO4</td>
<td></td>
</tr>
</tbody>
</table>
Programming using Java Lab

Course Objectives:
1. To Introduce object oriented concept for solving the real world problem
2. To introduce Exception Handling for detecting run time errors
3. To understand the utilization of multithreading concept
4. To build GUI Application using Swing ,AWT, Applet

Course Outcomes:
1. Execute the given problem using java language.
2. Construct the program using oops concept.
3. Demonstrate the usage of multithreading and exception handling mechanisms
4. Implement simple GUI interfaces for a computer program to interact with users, and the event-based GUI handling principles.

List of Experiments:
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 that prompts the user for an integer and then prints out all prime numbers up to that integer.
   B. Develop a java Program to multiply 2 given matrices.
3. A. Develop a java Program for sorting a given list of names in ascending order.
   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.
4. Develop a Java program to create an exception by name Minor Exception, which will be raised when the Age of the voter is less than 18 years.
5. 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.
6. Develop a Java program that correctly implements producer consumer problem using the concept of inter thread communication.
7. 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.
8. 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
9. Develop a Java program to demonstrate Mouse Listener, Mouse Motion Listener
10. 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.

11. Develop a java Program to design a calculator for implementing basic functions like +, *, -, / using grid layout.

12. Develop an applet that moves the character up, down, left and right when the appropriate arrows are pressed.

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>PO1 PO2 PO3 PO4 PO5 PO6 PO7 PO8 PO9 PO10 PO11 PO12</td>
<td>PSO1 PSO2 PSO3</td>
</tr>
<tr>
<td>CO1</td>
<td>3 1 2</td>
<td>1</td>
</tr>
<tr>
<td>CO2</td>
<td>3 1 2</td>
<td>1</td>
</tr>
<tr>
<td>CO3</td>
<td>3 1 3</td>
<td>1</td>
</tr>
<tr>
<td>CO4</td>
<td>3 1 3</td>
<td>1</td>
</tr>
</tbody>
</table>
Software Engineering Lab

Prerequisite: “Programming for Problem Solving”

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

Course Outcomes:
1. Able to prepare SRS document, design document, test cases and software configuration management and risk management related document.
2. Develop function oriented and object-oriented software design using tools like rational rose.
3. Able to perform unit testing and integration testing.
4. Apply various white box and black box testing techniques

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

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

Reference Books:

3. The unified modeling language user guide Grady Booch, James Rambaugh, Ivar Jacobson,Pearson Education.

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>CO1</td>
<td>H</td>
<td></td>
</tr>
<tr>
<td>CO2</td>
<td>H</td>
<td></td>
</tr>
<tr>
<td>CO3</td>
<td>H</td>
<td></td>
</tr>
<tr>
<td>CO4</td>
<td>H</td>
<td></td>
</tr>
</tbody>
</table>
DOING ENGINEERING-I

Pre-requisite: Basic Knowledge on Computers and C language

Course Objectives:
1) To Provide knowledge of different Smart System applications.
2) To provide knowledge of Arduino boards, Arduino as IDE, programming language, platform and basic components
3) Develop skills to design and implement various smart system application.
4) To know the basics of micro-controllers and sensors very quickly and can start building prototype.

Course Outcomes:
1) Able to Design Smart systems applications.
2) Understand about any new IDE, compiler, and MCU chip in Arduino compatible boards or similar types.
3) Understand Arduino programming with embedded C.
4) Learn how to prototype circuits with a breadboard

I-Microprocessor and Arduino
Introduction to Microprocessor, Micro controller, source mode and sink mode.
Arduino pin diagram, Types of Arduino boards and its pin description.

II-LED, LCD, Buzzer & Interfacing
Various types of Actuators, LED introduction, Buzzer introduction, Introduction to LCD 16x 2, Types of LCD’s , LCD Pin diagram ,Interfacing LED, Buzzer & LCD to Arduino.

LCD: Displaying name on LCD, Setting cursor position in LCD, Display and NO display in LCD, Scrolling Display towards left and right

LED: LED ON & OFF, TWO LED interfacing (Alternate LED ON & OFF), 1234 and 4321 LED pattern. LCD display with LED status.

Buzzer: Interfacing Buzzer with LED and LCD.

III-Motor driver and Interfacing to Arduino
L293D Pin diagram, Introduction to L293D Motor driver. Interfacing DC motors to Arduino, Designing robot

IV-Sensors and Programming
Introduction to soil moister sensor, gas sensor, Ultra Sonic sensor, Temperature sensor, LDR sensor, IR sensor and PIR sensor and its pin configurations.
1. Interfacing Soil moister sensor & gas sensor to Arduino
2. Interfacing LED, LCD and gas sensor
3. Interfacing Ultrasonic sensor to Arduino and displaying distance on screen
4. Interfacing Ultrasonic sensor to LED, LCD
5. Interfacing Temperature sensor to Arduino and Interfacing with LED.
6. Interfacing LDR sensor to Arduino and interfacing with LED.
7. Interfacing IR sensor to Arduino and interfacing with LED and buzzer.
8. Interfacing to PIR sensor to Arduino.
V-Bluetooth & RFID
Introduction, Principle, Bluetooth pin configuration, Two modes of Bluetooth, Interfacing Bluetooth to Arduino and LCD.
**RFID**: Introduction RFID, Principle, Interfacing to Arduino and LED.

**Expecting outcomes through assessments**: 

1: Designing Automatic Street Light  
2: Smart stick for blind person.  
3: Design Smart water tank

**TEXTBOOKS**:  

**REFERENCE BOOKS**:  

**CO-PO MAPPING**:

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