INSTITUTE OF AERONAUTICAL ENGINEERING
(Autonomous)
Dundigal - 500 043, Hyderabad, Telangana

COURSE CONTENT

PROGRAMMING FOR PROBLEM SOLVING LABORATORY								
II Semester: AE / ME / CE / ECE / EEE / CSE / IT / CSE (AI&ML) / CSE (DS)								
Course Code	**Category**	**Hours / Week**			**Credits**	**Maximum Marks**		
ACSE07	**Foundation**	**L**	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
		0	0	2	1	40	60	100
Contact Classes: Nil	**Tutorial Classes: Nil**	**Practical Classes: 45**				**Total Classes: 45**		
Prerequisite: There is no prerequisite to take this course								

I. COURSE OVERVIEW:
The course is designed with the fundamental programming skills and problem-solving strategies necessary to tackle a wide range of computational challenges. Through hands-on programming exercises and projects, students will learn how to write code, analyze problems and develop solutions using various programming languages and tools. The course will cover fundamental programming concepts and gradually progress to more advanced topics.

II. COURSES OBJECTIVES:
The students will try to learn

I	The fundamental programming constructs and use of collection data types in Python.
II	The ability to develop programs using object-oriented features.
III	Basic data structures and algorithms for efficient problem-solving.
IV	Principles of graph theory and be able to apply their knowledge to a wide range of practical problems across various disciplines.

III. COURSE OUTCOMES:
At the end of the course students should be able to:

CO1	Adapt programming concepts, syntax, and data structures through hands on coding exercises
CO2	Develop the ability to solve a variety of programming problems and algorithms using python
CO3	Implement complex and custom data structures to solve real-world problems.
CO4	Demonstrate proficiency in implementing graph algorithms to solve variety of problems and scenarios.
CO5	Develop critical thinking skills to solve the various real-world applications using graph theory.
CO6	Learn the importance of numerical methods and apply them to tackle a wide range of computational problems.

IV. COURSE CONTENT:

Week – 1: GETTING STARTED EXERCISES

Write Python programs for the following:

a. Add the two integer arrays and returning the indices
b. Find the duplicates in the integer array
c. Convert the Roman letters to integer form.
d. Find the Majority Element in the array.
e. Find the Number of Steps to Reduce a Number to Zero.

Week – 2: IMPLEMENTATION OF MATRX OPERATIONS

Write Python programs for the following:

a. Design and implementation of Matrix Addition and Matrix Multiplication
b. Design and implementation of Transpose of a Matrix
c. Design and implementation of Matrix Product.

Week – 3: IMPLEMENTATION OF STACK

Write Python programs for the following:

a. Design and implementation Stack and its operations using List
b. Uses Stack operations to implement Balanced Parenthesis Checking.
c. Uses Stack operations for evaluating the postfix expression.

Week – 4: IMPLEMENTATION OF QUEUE

Write Python programs for the following:

a. Design and implementation Queue and its operations using List
b. Design and implementation Queue using Stack
c. Design and implementation Circular Queue

Week – 5: IMPLEMENTATION OF GRAPHS

Write Python programs for the following:

a. Design and implementation of Number of Sink Nodes in a Graph Graphs
b. Design and implementation of Connected Components in a Graph
c. Design and implementation of Transpose of a Graph

Week – 6: IMPLEMENTATION OF GRAPH ALGORITHMS

Write Python programs for the following:

a. Design and implementation of Seven Bridges of Konigsberg
b. Design and implementation of Hamiltonian Cycle
c. Design and implementation of Number of Hamiltonian Cycle

Week –7: IMPLEMENTATION OF GRAPH ALGORITHMS

Write Python programs for the following:

a. Design and implementation of Travelling Salesman Problem
b. Design and implementation of Shortest Paths from Source to all Vertices (Dijkstra's Algorithm)
c. Design and implementation of Shortest Cycle in an Undirected Unweighted Graph
d. Design and implementation of Count Unique and all Possible Paths in a M x N Matrix

Week –8: IMPLEMENTATION OF GRAPH COLORING

Write Python programs for the following:

a. Design and implementation of Graph Coloring using Greedy Algorithm
b. Design and implementation of Coloring a Cycle Graph
c. Design and implementation of m Coloring Problem
d. Design and implementation of Edge Coloring of a Graph

Week –9: IMPLEMENTATION OF GRAPH TRAVERSALS

Write Python programs for the following:

a. Design and implementation of Breadth First Search
b. Design and implementation of Depth First Search

Week –10: IMPLEMENTATION OF MINIMUM SPANNING TREE (MST)
Write Python programs for the following:
a. Design and implementation of Kruskal's Algorithm
b. Design and implementation of Prim's Algorithm

Week –11: IMPLEMENTATION OF ROOTS OF EQUATIONS
Write Python programs for the following:
a. Design and implementation of Bisection Method
b. Design and implementation of Method of False Position
c. Design and implementation of Newton Raphson Method
d. Design and implementation of Secant Method and Muller Method

Week –12: IMPLEMENTATION OF NUMERICAL INTEGRATION
Write Python programs for the following:
a. Design and implementation of Trapezoidal Rule for Approximate Value of Definite Integral
b. Design and implementation of Simpson's 1/3 Rule Algorithm
c. Design and implementation of Simpson's 3/8 Rule

Week –13: IMPLEMENTATION OF ORDINARY DIFFERENTIAL EQUATIONS
Write Python programs for the following:
a. Design and implementation of The Euler Method
b. Design and implementation of Runge-Kutta Second Order Method

V. TEXT BOOKS:

1. Eric Matthes, "Python Crash Course: A Hands-On, Project-based Introduction to Programming", No Starch Press, 3rd Edition, 2023.
2. John M Zelle, "Python Programming: An Introduction to Computer Science", Ingram short title, 3rd Edition, 2016.

VI. REFERENCE BOOKS:

1. Yashavant Kanetkar, Aditya Kanetkar, "Let Us Python", BPB Publications, 2nd Edition, 2019.
2. Martin C. Brown, "Python: The Complete Reference", Mc. Graw Hill, Indian Edition, 2018.
3. Paul Barry, "Head First Python: A Brain-Friendly Guide", O'Reilly, 2nd Edition, 2016
4. Taneja Sheetal, Kumar Naveen, "Python Programming – A Modular Approach", Pearson, 1st Edition, 2017.
5. R Nageswar Rao, "Core Python Programming", Dreamtech Press, 2018.

VII. ELECTRONICS RESOURCES
1. https://realPython.com/Python3-object-oriented-programming/
2. https://Python.swaroopch.com/oop.html
3. https://Python-textbok.readthedocs.io/en/1.0/Object_Oriented_Programming.html
4. https://www.programiz.com/Python-programming/
5. https://www.geeksforgeeks.org/python-programming-language/

VIII. MATERIALS ONLINE
1. Course template
2. Lab Manual