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

COURSE CONTENT

DATA STRUCTURES LABORATORY								
II Semester: AE / ME / CE / ECE / EEE / CSE / IT / CSE (AI&ML) / CSE (DS)								
Course Code	**Category**	**Hours / Week**			**Credits**	**Maximum Marks**		
ACSE06	**Foundation**	**L**	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
		-	-	2	1	40	60	100
Contact Classes: Nil	**Tutorial Classes: Nil**	**Practical Classes: 45**				**Total Classes: 45**		
Prerequisite: **Essentials of Problem Solving**								

I. COURSE OVERVIEW:

The course covers some of the general-purpose data structures and algorithms, and software development. Topics covered include managing complexity, analysis, static data structures, dynamic data structures and hashing mechanisms. The main objective of the course is to teach the students how to select and design data structures andalgorithms that are appropriate for problems that they might encounter in real life. This course reaches to studentby power point presentations, lecture notes, and lab which involve the problem solving in mathematical and engineering areas.

II. COURSES OBJECTIVES:

The students will try to learn

I. To provide students with skills needed to understand and analyze performance trade-offs of different algorithms / implementations and asymptotic analysis of their running time and memory usage.
II. To provide knowledge of basic abstract data types (ADT) and associated algorithms: stacks, queues, lists, tree, graphs, hashing and sorting, selection and searching.
III. The fundamentals of how to store, retrieve, and process data efficiently.
IV. To provide practice by specifying and implementing these data structures and algorithms in Python.
V. Understand essential for future programming and software engineering courses.

III. COURSE OUTCOMES:

At the end of the course students should be able to:

CO 1 Interpret the complexity of algorithm using the asymptotic notations.

CO 2 Select appropriate searching and sorting technique for a given problem.

CO 3 Construct programs on performing operations on linear and nonlinear data structures for organization of a data.

CO 4 Make use of linear data structures and nonlinear data structures solving real time applications.

CO 5 Describe hashing techniques and collision resolution methods for efficiently accessing data with respect to performance.

CO 6 Compare various types of data structures; in terms of implementation, operations and performance.

IV. LIST OF EXPERIMENTS:

WEEK-1: GETTING STARTED EXERCISES

a. Recursively Remove all Adjacent Duplicates
b. Product of Two Numbers using Recursion
c. Binary to Gray Code using Recursion
d. Fibonacci Series in Reverse Order using Recursion

WEEK-2: SEARCHING TECHNIQUES

Write Python programs for implementing the following searching techniques.
a. Linear search.
b. Binary search.
c. Uniform Binary Search
d. Interpolation Search
e. Fibonacci search.

WEEK-3: SORTING TECHNIQUES

Write Python programs for implementing the following sorting techniques to arrange a list of integers in ascending order.
a. Bubble sort.
b. Insertion sort
c. Selection sort.

WEEK-4: SORTING TECHNIQUES

Write Python programs for implementing the following sorting techniques to arrange a list of integers in ascending order.
a. Quick sort.
b. Merge sort
c. Heap sort
d. Radix sort
e. Shell sort

WEEK-5: IMPLEMENTATION OF STACK

Write Python programsto
a. Design and implementation Stack and its operations using Arrays.
b. Uses Stack operations to convert infix expression into postfix expression.
c. Uses Stack operations for evaluating the postfix expression
d. Implementation of reversing a stack

WEEK-6: IMPLEMENTATION OF QUEUE

Write Python programs for the following:
a. Design and implementation Queue and its operations using Arrays
b. Implementation of Queue using Stack
c. Implementation of Circular Queue
d. Implementation of Double Ended Queue

WEEK-7: IMPLEMENTATION OF SINGLE LINKED LIST

Write Python programs for the following:
a. Uses functions to perform the following operations on single linked list.
(i) Creation (ii) insertion (iii) deletion (iv) traversal
b. To store a polynomial expression in memory using linked list
c. Find middle element of a single linked list

WEEK-8: IMPLEMENTATION OF CIRCULAR SINGLE LINKED LIST

Write Python programs for the following:
Uses functions to perform the following operations on Circular linked list.
(i) Creation (ii) insertion (iii) deletion (iv) traversal

WEEK-9: IMPLEMENTATION OF DOUBLE LINKED LIST

Write Python programs for the following:
a. Uses functionsto perform the following operations on double linked list.

(i) Creation (ii) insertion (iii) deletion (iv) traversal in both ways.

b. Implementation to delete all occurrences of a given key

WEEK-10: IMPLEMENTATION OF STACK, QUEUE USING LINKED LIST

a. Write Python programs to implement stack using linked list.
b. Write Python programs to implement queue using linked list

WEEK-11: IMPLEMENTATION OF AVL TREES

a. Insertion
b. Deletion
c. Count greater nodes in AVL trees

WEEK-12: GRAPH TRAVERSAL TECHNIQUES

Write Python programs to implement the following graph traversal algorithms:

a. Depth first search.
b. Breadth firstsearch.

WEEK-13: IMPLEMENTATION OF BINARY TREE

Write a Python program that uses functionsto perform the following:
a. Create a binary tree.
b. Insert, delete a node in a binary tree

WEEK-14: IMPLEMENTATION OF BINARY SEARCH TREE

Write a Python program that uses functionsto perform the following:
c. Create a binary search tree.
d. Traverse the above binary search tree recursively in pre-order, post-order, and in-order.
e. Count the number of nodes in the binary search tree.

V. TEXT BOOKS:

1. Yashavant Kanetkar, Aditya Kanetkar, “Let us Python”, BPB publication, 1st Edition, 2019.
2. Ashok Kamthane, Amit Kamthane, “Programming and Problem Solving with Python”, McGraw Hill Education (India) Private Limited, 2018.

VI. REFERENCE BOOKS:

1. Taneja Sheetal, Kumar Naveen, “Python Programming – A modular approach”, Pearson, 2017.
2. Michael H Goldwasser, David Letscher, “Object Oriented Programming in Python”, Prentice Hall, 1st edition, 2007.

VII. ELECTRONICS RESOURCES:

1. https://python-textbok.readthedocs.io/en/1.0/Object_Oriented_Programming.html
2. https://www.programiz.com/python-programming/

VIII. MATERIALS ONLINE

1. Course template
2. Lab Manual