# 20EID237: DATA STRUCTURES WITH PYTHON

| L | T | P | C |
|---|---|---|---|
| 2 | 1 | 3 | 3.5 |

The study of data structures, a fundamental component of a computer science education, serves as the foundation upon which many other computer science applications are built. Knowledge of data structures is a must for students who wish to work in design, implementation, testing or maintenance of any software system. Organization of data in an efficient way for application is the major focus of the course.

**Course Objectives**

- Introduce various data representation methods and searching methods.
- Familiarize with linear data structures and operations on them.
- Demonstrate the organization of data as trees and various operations on trees.
- Teach various graph representations.
- Enable to perform graph traversal and find shortest path and minimal spanning tree for a graph
- Expose common sorting techniques and their complexities.

**COURSE CONTENTS**

**UNIT I** **12 L**

**Python Primitives:** Python overview, Objects in Python, Expressions, Operators and Precedence, Control Flow, Functions, Simple Input and Output, Exception handling, Iterators and Generators, Collections [Strings, Lists, Tuples, Dictionaries].

**Learning Outcomes:**

After completion of this unit, the student will be able to

- summarize various ways of representing data (L2)
- explain the working of linear and binary search algorithms (L2)
- compare various data representations and search algorithms (L6)

**UNIT II** **10L**

**Algorithm Analysis:** Asymptotic Analysis and Big O notation Recursion: What is recursion, examples [Factorial functions, Fibonacci series]. **Array Based Sequences:** Python Sequence types, low-level arrays, dynamic arrays, efficiency of python's sequences, using array-based sequences. **Searching:** Sequential Search, binary search and algorithmic analysis. **Sorting:** Insertion sort, selection sort, bubble sort

**Learning Outcomes:**

After completion of this unit, the student will be able to

- summarize various ways of representing data (L2)
- explain the working of linear and binary search algorithms (L2)
- compare data representations and sorting algorithms (L6)

**UNIT III** **12 L**

**Sorting:** quick sort, merge sort and their algorithmic analysis. **Linked lists:** Single linked list, double linked list, circular linked list **Stacks:** Definition, operations: array implementation, linked list implementation. **Queues:** Definition, operations: array implementation, linked list implementation and applications, Priority Queue. Double-Ended Queues.

**Learning Outcomes:**

After completion of this unit, the student will be able to

- discuss how stacks and queues are implemented using arrays and linked lists (L2)
- explain the implementation of priority queues (L2)
- list the applications of stacks, queues and priority queues (L1)
- compare different types of linked lists (L6)

**UNIT IV** **12 L**

**Trees:** Definition, Tree properties, **Binary trees:** properties, implementation, tree traversals, Heap tree, Heap sort Search Trees: binary search tree, AVL trees and operations on AVL trees, and (2, 3)-Trees

**Learning Outcomes:**

After completion of this unit, the student will be able to

- discussthe properties of trees, binary, binary search and AVL trees (L2)
- explain how operations such as insertion, deletion and traversal are performed on different types of trees (L2)
- analyze the complexity of operations on different tree types (L4)

**UNIT V** **10 L + 9 P**

**Graphs:** ADT, data structure for graphs, graph traversal, Transitive closure, directed acyclic graph, shortest paths [weighted graphs, dijkstra's algorithm], minimum spanning trees [Prim's, Kruskal's, disjoint partitions, union-find structures].

**Learning Outcomes:**

After completion of this unit, the student will be able to

- demonstrate different graph representations and operations (L2)
- discuss the working of common sorting algorithms (L2)
- analyze the computational efficiency of algorithms for sorting (L4)

**Text Book(s):**

1. Michel T. Goodrich, Roberto Tamassia, Michel H. Goldwasser, Data Structures & Algorithms in Python, Willey March 2013. ISBN: 978-1-118-29027-9
2. Rance D. Necaise, Data Structures & Algorithms using Python, John Willey & Sons, India. ISBN 9788126562169

**References:**

1. Wesly J. Chun, Core Python Programming, Prenctice Hall, 2/e,
2. Mastering Machine Learning with Six Steps by Manohar Swamynathan , Apress, ISBN-13: 978-1-4842-2866-1
3. Python for Probability, Statistics, and Machine Learning by José Unpingco, Springer ISBN 978-3-319-30717-6 (eBook)
4. Python Programming using problem solving Approach by ReemaThareja, Oxford University, Higher Education Oxford University Press; First edition (10 June 2017), ISBN-10: 0199480173.
5. Introduction to Computation and Programming Using Python by John V Guttag, — Revised and expanded Edition, MIT Press , 2013
6. Fundamentals of Python first Programmes by Kenneth A Lambert, Copyrighted material
Course Technology Inc. 1st edition (6th February 2009).
7. Algorithmic Problem Solving with Python by John B. Schneider Shira Lynn Broschat Jess Dahmen,

**Course Outcomes:**

After Completion of this course, the student will be able to:

- explain various ways of representing data in a computer (L2)
- demonstrate operations on linear data structures (L2)
- discuss the mechanisms for creating, altering and traversing various types of trees (L2)
- explain the representations, traversals and applications of graphs (L2)
- analyze common sorting algorithms (L4)
- choose a data structure that gives the best performance for a given application (L6)

# DATA STRUCTURES WITH PYTHON LABORATORY

This Lab provides hands-on experience in designing, implementing, and using the most-commonly used data structures including arrays, stacks, queues, linked lists, trees, hash tables and graphs.

Implementation different searching and sorting algorithms is also done.

List of Practical Experiments:

1. Python sample programs for practice
   - Find minimum among three numbers.
   - Find the GCD and LCM of two/three numbers
   - Check whether the given number is perfect
   - Print Twin Primes up to a Specified limit.
   - Print the prime numbers up to a specified limit.
   - Find the sum of digits of a number. Check whether given number is Armstrong number or not.
   - Swap of Two numbers
   - Performs all the five arithmetic operations.
2. Write 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.
3. Write programs for:
   - Representing sparse matrix
   - Sparse matrix addition
   - Sparse matrix transpose
4. Write a program to Perform Linear Search and Binary Search on a list stored in an array.
5. Write 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. Write a program to create a doubly 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
7. Write a program to create a Circular singly linked list for adding and deleting a Node.
8. Write a program to create a stack and perform various operations on it.
9. Write a program to convert the infix expression into postfix form.
10. Write a program to create a queue and perform various operations on it.
11. Write a program to create a binary tree and perform various traversals.
12. Write a program to create a binary search tree and perform search operation.
13. Write a program to implement Depth First Search, Breadth First Search traversals on a graph.
14. Write a program to implement Dijkstra’s Shortest Path Algorithm
15. Write a program to implement various sorting techniques: [Compare with Python’s Built-In Sorting Functions also]
    - Insertion sort
    - Selection Sort
    - Bubble Sort
    - Merge Sort
    - Quick Sort