Teaching Plan

Course: B.Sc.(Physical Sciences)

Semester-II

Subject: DSC02: Data Structures using C++

Learning Objectives

The course aims at developing the ability to use basic data structures like arrays, stacks, queues, lists, trees to solve problems. C++ is chosen as the language to understand implementation of these data structures.

Learning outcomes

On successful completion of the course, students will be able to:

- Compare two functions for their rates of growth.
- Understand abstract specification of data-structures and their implementation.
- Compute time and space complexity of operations on a data-structure.
- Identify the appropriate data structure(s) for a given application and understand the trade-offs involved in terms of time and space complexity.
- Apply recursive techniques to solve problems.

Week	Topic
Week 1	**UNIT – I** Functions used in analysis, asymptotic notations
Week 2	**UNIT – I** Asymptotic analysis, solving recurrences using recursion tree, Master Theorem
Week 3	**UNIT-II** Arrays: array operations, applications, sorting, two-dimensional arrays, Dynamic allocation of arrays;
Week 4	**UNIT-II**

	Linked Lists: singly linked lists, doubly linked lists, Circularly linked lists, **(Assignment-1)**
Week 5	**UNIT-II** Stacks: stack as an ADT, implementing stacks using arrays, implementing stacks using linked lists, applications of stacks;
Week 6	**UNIT-II** Queues: queue as an ADT, implementing queues using arrays, implementing queues using linked lists, double-ended queue as an ADT, Time complexity analysis of operations on all data structures **(Test-1)**
Week 7	**UNIT – III** Sorting: Insertion Sort, Count Sort and their complexity analysis
Week 8	**UNIT – IV** Recursion: Recursive functions, linear recursion, binary recursion. **(Assignment-2)**
Week 9	**UNIT – V** Trees, Binary Trees. Trees: definition and properties, binary trees: definition and properties
Week 10	**UNIT – V** Traversal of binary trees and their time complexity analysis.
Week 11	**UNIT – VI** Binary Search Trees, Balanced Search Trees: Binary Search Trees: insert, delete (by copying), search operations
Week 12	**UNIT – VI** Time complexity analysis of tree operations; Balanced Search Trees **(Test-2)**
Week 13	**UNIT – VI** (2,4) Trees:
Week 14	**UNIT – VII** Binary Heap, Priority Queue: Binary Heaps: motivation and introduction, Application of heaps - Priority Queues and Revision

Practical component

1. Perform matrix addition and multiplication.

2. Implement following recursive functions:

a. Factorial of a number

b. N^{th} fibonacci number

c. Power function: x^y

3. Implement singly linked lists.

4. Implement doubly linked lists.

5. Implement circular linked lists.

6. Implement stack data structure and its operations using arrays.

7. Implement stack data structure and its operations using linked lists.

8. Convert Prefix expression to Infix and Postfix expressions, and evaluate.

9. Implement queue data structure and its operations using arrays.

10. Implement queue data structure and its operations using linked lists.

11. Implement Binary Trees and its traversals.

and additional programs as per the syllabus

Essential/recommended readings

1. Goodrich, M., Tamassia, R., & Mount, D., Data Structures and Algorithms Analysis in C++, 2nd edition. Wiley, 2011.

2. Cormen, T.H., Leiserson, C.E., Rivest, R. L., Stein C., Introduction to Algorithms, 3rd edition, Prentice Hall of India, 2010.

3. Drozdek, A., Data Structures and Algorithms in C++, 4th edition, Cengage Learning, 2012.

Suggestive readings

1. Sahni, S. Data Structures, Algorithms and applications in C++. 2nd Edition. Universities Press, 2011.

2. Tanenbaum, A. M., Augenstein, M. J., & Langsam Y., Data Structures Using C and C++. 2nd edition. Prentice Hall of India, 2009.