

**BSc. (Physical Sciences/ Mathematical Sciences) with Computer Science as one of the Core Disciplines**

**Category II**

**DISCIPLINE SPECIFIC CORE COURSE (DSC-2): Data Structures using C++**

**CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE**

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
DSC02: Data Structures using C++	4	3	0	1	Class pass with Mathematics XII	Nil

**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.

**SYLLABUS OF DSC-2 UNIT – I (06 Hours)**

**Growth of Functions, Recurrence Relations.** Functions used in analysis, asymptotic notations, asymptotic analysis, solving recurrences using recursion tree, Master Theorem.

**UNIT – II (12 Hours)**

**Arrays, Linked Lists, Stacks, Queues, Deques.** Arrays: array operations, applications, sorting, two-dimensional arrays, dynamic allocation of arrays; Linked Lists: singly linked lists, doubly linked lists, circularly linked lists, Stacks: stack as an ADT, implementing stacks using arrays, implementing stacks using linked lists, applications of stacks; 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.

### **UNIT – III (06 Hours)**

**Sorting:** Insertion Sort, Count Sort and their complexity analysis.

### **UNIT – IV (03 Hours)**

**Recursion:** Recursive functions, linear recursion, binary recursion.

### **UNIT – V (06 Hours)**

**Trees, Binary Trees.** Trees: definition and properties, binary trees: definition and properties, traversal of binary trees and their time complexity analysis.

### **UNIT – VI (09 Hours)**

**Binary Search Trees, Balanced Search Trees:** Binary Search Trees: insert, delete (by copying), search operations, time complexity analysis of these operations; Balanced Search Trees and (2,4) Trees: motivation and introduction.

### **UNIT – VII (03 Hours)**

**Binary Heap, Priority Queue:** Binary Heaps: motivation and introduction, application of heaps - Priority Queues.

### **Practical component (if any) – 30 Hours**

1. Perform matrix addition and multiplication.
2. Implement following recursive functions:
  - a. Factorial of a number
  - b.  $N^{\text{th}}$  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.

### **Essential/recommended readings**

1. Goodrich, M., Tamassia, R., & Mount, D., *Data Structures and Algorithms Analysis in C++*, 2<sup>nd</sup> edition. Wiley, 2011.
2. Cormen, T.H., Leiserson, C.E., Rivest, R. L., Stein C., *Introduction to Algorithms*, 3<sup>rd</sup> edition, Prentice Hall of India, 2010.
3. Drozdek, A., *Data Structures and Algorithms in C++*, 4<sup>th</sup> edition, Cengage Learning, 2012.

### **Suggestive readings**

- (i) Sahni, S. *Data Structures, Algorithms and applications in C++*. 2<sup>nd</sup> Edition. Universities Press, 2011.
- (ii) Tanenbaum, A. M., Augenstein, M. J., & Langsam Y., *Data Structures Using C and C++*. 2<sup>nd</sup> edition. Prentice Hall of India, 2009.