

Course 01 | Basic Data Structures | 03 Credit and

Course 02 | Problem Solving (Part 1) | 1.5 Credit

Week 1: Introduction to C++

- **Module 01: Introduction to C++ Language - Basics**

- 1-1: Introduction

- 1-2: How to print in C++

- 1-3: How to take input in C++

- 1-4: Namespaces in C++

- 1-5: If else in C++

- 1-6: Loop in C++

- **Module 02: More about C++**

- 2-1: Array and String in C++

- 2-2: Function in C++

- 2-3: Pointers in C++

- 2-4: Dynamic Allocation in C++

- 2-5: Vector in C++

- 2-6: swap(), min(), max(), sort() functions in C++

- **Module 2.5: Week 01 Practice Day 01**

- **Module 03: Lab Class: C++ Class**

- 3-1: Class and Object in C++

- 3-2: Access modifiers

- 3-3: Constructor and Destructor

- 3-4: Dynamic Object in C++

- 3-5: Array of Class

- 3-6: Sort Array of Class

- **Module 3.5: Week 01 Practice Day 02**

Week 2: Time Complexity and Array Operations

- **Module 4: Time Complexity and Linear Search**

- 4-1: Factors of Judging Algorithms

- 4-2: Time Complexity Part 1

- 4-3: Time Complexity Part 2

- 4-4: Time Complexity Part 3

- 4-5: Searching - Linear Search

- 4-6: Linear Search Code Implementation

- **Module 5: Array Operations**

- 5-1: Searching - Binary Search

- 5-2: Binary Search Code Implementation

- 5-3: Array Insertion

- 5-4: Array Insertion Code Implementation

- 5-5: Array Deletion

- 5-6: Array Deletion Code Implementation

- **Module 5.5: Week 02 Practice Day 01**

- **Module 6: Lab Class: Application of Array (Sorting)**

- 6-1: Introduction to Sorting

- 6-2: Bubble Sort Part 1

- 6-3: Bubble Sort Part 2

- 6-4: Bubble Sort Part 3

- 6-5: Insertion Sort Part 1

- 6-6: Insertion Sort Part 2

- **Module 6.5: Week 02 Practice Day 02**

- **Module 7: Lab Assignment 01**

Week 3: Merge Sort and Introduction to Linked List

- **Module 8 Merge Sort**

- 8-1 Why Merge Sort

- 8-2 Merge Sort Theory

- 8-3 Merge Sort Complexity Analysis

- 8-4 Merge Sort Implementation

- 8-5 Quick Sort

- 8-6 Summary

- **Module 9 Introduction to Linked List**

- 9-1 Quick Sort Implementation

- 9-2 Quick Sort Complexity

- 9-3 Linear Linked List – Concept Part 1

- 9-4 Linear Linked List – Concept Part 2

- 9-5 Linear Linked List – Concept Part 3

- **Module 9.5: Week 03 Practice Day 01**

- **Module 10 Introduction to Linked List**

- 10-1 Linked List Implementation – Structure

- 10-2 Linked List Implementation – Insertion at head

- 10-3 Linked List Implementation – Traverse

- 10-4 Linked List Implementation – Search for a distinct value

- 10-5 Linked List Implementation – Search for all possible values

- **Module 10.5: Week 03 Practice Day 02**

- **Module 11: Theory Assignment 02**

Week 4: Linked List Operations

- **Module 12: Linear Linked List Operations 1**

- 12-1 Recap of Linear Linked List

- 12-2 Length of a Linear Linked List

- 12-3 Linked List Implementation – Insert at any position

- 12-4 Linked List Implementation – Deletion at Head

- 12-5 Linked List Implementation – Deletion at any position

- **Module 13: Linear Linked List Operations 2**

- 13-1 Insertion after a specific value

- 13-2 Maintain tail of a Linked List

- 13-3 Deletion at Tail

- 13-4 Finding the midpoint of a Linear Linked List

- 13-5 Reverse Print of a Linear Linked List.

- **Module 13.5: Week 04 Practice Day 01**

- **Module 14: Linked List Advanced Operations | Part 1**

- 14-1 Reverse a Linear Linked List (Non-Recursive)

- 14-2 Reverse a Linear Linked List (Recursive)

- 14-3 Doubly Linked List Part 1

- 14-4 Doubly Linked List Part 2

- **Module 14.5: Week 04 Practice Day 02**

- **Module 15: Exercise Day**

Week 5: Linked List Application | Stack, Queue

- **Module 16: Introduction to Stack**

- 16-1 Introduction to Stack
- 16-2 Stack using Array Theory
- 16-3 Stack using Array Implementation
- 16-4 Stack using LinkedList Theory
- 16-5 Stack using LinkedList Implementation

- **Module 17: STL Stack, Linked List and Application of Stack**

- 17-1 list in C++
- 17-2 stack in C++
- 17-3 Stack | Problem 1
- 17-4 Stack | Problem 2
- 17-5 Stack | Problem 3

- **Module 17.5: Week 05 Practice Day 01**

- **Module 18: Application of Stack Cont. and Intro to Queue**

- 18-1 Stack | Problem 4
- 18-2 Stack | Problem 5
- 18-3 Introduction to Queue
- 18-4 Queue using LinkedList Implementation
- 18-5 queue in C++

- **Module 18.5: Week 05 Practice Day 02**

- **Module 19: Lab Mid Term**

Week 6: Linked List Application | Queue, Dequeue

- **Module 20: Application of Queue**

20-1 Queue | Problem 1

20-2 Queue | Problem 2

20-3 Queue | Problem 3

20-4 Queue | Problem 4

20-5 Queue | Problem 5

- **Module 21: Introduction to Dequeue**

21-1 Introduction to Dequeue

22-2 Dequeue implementation using Linked List

23-3 dequeue in C++

24-4 Dequeue | Problem 1

25-5 Dequeue | Problem 2

- **Module 21.5**
- **Module 22: Lab 6: Practice Problems**
- **Module 22.5**
- **Module 23: Exam 05**

Week 7: Binary Tree

- **Module 24: Introduction to Graph and Tree**

24-1 Introduction to Non-Linear Data Structure

24-2 Introduction to Graph

24-3 Introduction to Tree and Binary Tree

24-4 Variants of Binary Tree

24-5 Binary Tree Implementation Part 1 (Structure)

- **Module 25: Binary Tree**

25-1 Binary Tree Implementation| Part-2 (Print Tree BFS)

25-2 Binary Tree Implementation| Part-3 (Print Tree DFS)

25-3 Binary Tree Implementation | Inorder, Preorder, and Postorder Traversal

25-4 Introduction to Binary Search Tree| Part-1

25-5 Introduction to Binary Search Tree| Part-2

- **Module 25.5: Practice Day**
- **Module 26: Binary Search Tree**
- **Module 26.5: Practice Day 02**
- **Module 27: Exam 06**

Week 8: Heap and Priority Queue

- **Module 28: Introduction to Heap**

28-1 Introduction to Heap

28-2 Heap Implementation Part 1

28-3 Heap Implementation Part 2

28-4 Heap Implementation Part 3

- **Module 29: Heap Sort and Priority Queue**

29-1 Heap Sort Theory

29-2 Heap Sort Implementation

29-3 Priority Queue

29-4 Priority Queue Application| Problem 1

29-5 Priority Queue Application| Problem 2

- **Module 29.5: Practice Day**
- **Module 30: Priority Queue Problems**
- **Module 30.5: Practice Day 02**
- **Module 31: Theory Final Exam**

Week 9: Disjoint Set Union, Trie and STL

- **Module 32: STL**
- **Module 33: Disjoint Set Union**
- **Module 33.5: Practice Day**
- **Module 34: Trie**
- **Module 34.5: Practice Day 02**
- **Module 35: Lab Final Exam**

