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 Introduction
  13-2 Linked LIst Insert after a value
  13-3 Reverse Print of a Linked List
  13-4 Intro to Doubly Linked List
  13-5 Doubly Linked List Insertion
  13-6 Doubly Linked List Deletion
  13-7 Summary
- **Module 13.5: Week 04 Practice Day 01**
- **Module 14: Doubly Linked List | Part 1**

  14-1 Doubly Linked List Structure
  14-2 Doubly Linked List Insertion At Head
  14-3 Doubly Linked List Insertion At any index
  14-4 Doubly Linked List Deletion
  14-5 Doubly Linked List Reverse
  14-6 Summary
- **Module 14.5: Week 04 Practice Day 02**
- **Module 15: Theory MidTerm**

Week 5: Recap Week

- **Module 16: Time Complexity**

  16-1 Introduction
  16-2 Complexity Theory
  16-3 Complexity Part 1
  16-4 Complexity Part 2
  16-5 Complexity Part 3
  16-6 Complexity Part 4
  16-7 Summary
- **Module 17: Recursion and Merge Sort**

  17-1 Recursion Part 1
  17-2 Recursion Part 2
  17-3 Recursion Part 3
  17-4 Merge Sort Part 1
  17-5 Merge Sort Part 2
  17-6 Merge Sort Complexity
  17-7 Summary
- **Module 17.5: Week 05 Practice Day 01**
- **Module 18: Linked List Recap**

  18-1 Linked List Overview
  18-2 Singly Linked List Insertion 1
  18-3 Singly Linked List Insertion 2
  18-4 Doubly Linked List Insertion 1
  18-5 Doubly Linked List Insertion 2
  18-6 Summary
- **Module 18.5: Week 05 Practice Day 02**
- **Module 19: Lab MidTerm**

Week 6: Linked List Application | Stack

- **Module 20: Introduction to Stack**

  20-1 Introduction to Stack
  20-2 Stack using Array Theory
  20-3 Stack using Array Implementation
  20-4 Stack using LinkedList Theory
  20-5 Stack using LinkedList Implementation
  20-6 Summary
- **Module 21: C++ Template and Stack**

  21-1 What is Template in C++
  21-2 Stack Implementation using Template | Part 1
  21-3 Stack Implementation using Template | Part 2
  21-4 Reverse a stack using another stack
  21-5 Sort a stack using another stack
  21-6 Summary
- **Module 21.5: Week 06 Practice Day 01**
- **Module 22: STL Stack, Linked List and Application of Stack**

  22-1 list in C++ Part 1
  22-2 list in C++ Part 2
  22-3 stack in C++
  22-4 Regular Bracket Sequence Check
  22-5 Infix expression to Postfix expression
  22-6 Summary
- **Module 22.5: Week 06 Practice Day 02**
- **Module 23: Theory Assignment 02**

Week 7: Linked List Application | Queue, Deque

- **Module 24: Introduction to Queue**

  24-1 Introduction to Queue
  24-2 Queue using Array Theory
  24-3 Queue using Array Implementation
  24-4 Queue using Circular Array Theory
  24-5 Queue using Circular Array Implementation
  24-6 Dynamic Queue Implementation
  24-7 Summary
- **Module 25: Queue**

  25-1 Queue using linked list theory
  25-2 Queue using linked list implementation
  25-3 STL Queue
  25-4 Generate Binary numbers using Queue Theory
  25-5 Generate Binary numbers using Queue Implementation
  25-6 Summary
- **Module 25.5: Practice Day**
- **Module 26: Deque**

  26-1 Introduction to Deque
  26-2 Deque using Linked List Theory
  26-3 Deque using Linked List Implementation Part 1
  26-4 Deque using Linked List Implementation Part 2
  26-5 Reverse a Deque
  26-6 STL Deque
  26-7 Summary
- **Module 26.5: Practice Day**
- **Module 27: Lab Assignment 02**

Week 8: 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 9: 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 Heapsort 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: C++ STL**
- **Module 30.5: Practice Day 02**
- **Module 31: Theory Final Exam**

Week Number	Exam
Week 01	Module 7: Theory Assignment 01
Week 02	Module 11: Lab Assignment 01
Week 03	Module 15: Theory Mid-Term
Week 04	Module 19: Lab Mid-Term
Week 05	Module 23: Theory Assignment 02
Week 06	Module 27: Lab Assignment 02
Week 07	Module 31: Theory Final Exam
Week 08	Module 35: Lab Final Exam