Lecture 01: Introduction.

Basics:
- C and C++ Programming Language

Introduction to C++ Programming Language
- Setting up C++ Development Environment

C++ Programming Basics
- C++ Data Types
- Variables in C++
- Loops in C and C++
- Decision Making in C / C++ (if , if..else, Nested if, if-else-if)
- C Math Function

Lecture 02: Input/Output.

Basic Input/Output
- I/O Redirection in C and C++
- Basic Input / Output in C and C++
- Clearing The Input Buffer In C/C++

Lecture 03: Operators.

Operators
- Arrays
- Strings
- Functions
- Pointers & References

Lecture 04: BASIC CONCEPTS OF OBJECTS ORIENTED PROGRAMMING.

Object Oriented Programming
- Object Oriented Programming in C++
- C++ Classes and Objects

Inheritance in C++
- SubClass and SuperClass
- Use of Final KeyWord

Polymorphism in C++
- Difference btw. Inherittance and Polymorphism
- Use of Polymorphism
- Examples of different types
- Inner Classes
- External Classes
- Access of outer classes from inner class

Encapsulation in C++

Abstraction in C++

- Methods
- Classes
- Interfaces
- Why we use this so.?

Lecture 05: Overloading.

C++ Function
- Create Functions
- Use of Parentheses
- Call a Function

Function Declaration and Definition
- Declaration
- Definition

C++ Function Parameters
- Default Parameter
- Multiple Parameter
- Return Values
- Pass By Referance
- Recursion

Function Overloading in C++
- For Int and Double both

Operator Overloading in C++
- Use of operators in overloading

Lecture 06: Constructors & Destructors.

Constructors & Destructors
- How to create.
- Use of it.
- Examples

Constructors in C++
- How to create.
- Use of it.
- Examples

Default Constructors in C++
- How to create.
- Use of it.
- Examples

Copy Constructor in C++
- How to create.
- Use of it.
- Examples

Destructors in C++
- How to create.
- Use of it.
- Examples

Default Constructors in C++
- How to create.
- Use of it.
- Examples

Private Destructor in C++
- Why it is used ?
- How to create.
- Use of it.
- Examples

Lecture 07: Exception Handling.

Exception Handling
- How to create.
- Use of it.
- Examples

Exception Handling in C++
- How to create.
- Use of it.
- Examples

Lecture 08

- Control Structures
- Static Data Member
- Friendly functions
- Returning Objects
- Array reference out of bound
- Containership in C++

Lecture 09: Symbolic Constant

- Template
- Class Template
- Virtual destructors
- Managing Console I/O
- Namespaces
- New & Delete Operators
- Pure Virtual Functions

Lecture 10: Topics (Data Structure)

Briefing:
- **Array**
- **Linked List**
- **Stack**
- **Queue**
- **Binary Tree**

Lecture 11: Linked List.

Linked List:

Introduction to Linked List
- Linked List vs Array

Linked List Insertion
- Linked List Deletion (Deleting a given key)
- Linked List Deletion (Deleting a key at given position)

Approach of looking at Array vs. Linked List
- Find Length of a Linked List (Iterative and Recursive)
- How to write C functions that modify head pointer of a Linked List?
- Swap nodes in a linked list without swapping data

Lecture 12: Circular Linked List.

Circular Linked List:
- Circular Linked List Introduction and Applications,
- Circular Singly Linked List Insertion
- Circular Linked List Traversal
- Split a Circular Linked List into two halves
- Sorted insert for circular linked list
-

Lecture 13: Stack.

Stack:

Introduction to Stack
- Infix to Postfix Conversion using Stack.
- Evaluation of Postfix Expression.
- Reverse a String using Stack.
- Implement two stacks in an array.
- Check for balanced parentheses in an expression.
- Next Greater Element.

Lecture 14: Queue.

Queue:
- Queue Introduction and Array Implementation
- Linked List Implementation of Queue
- Applications of Queue Data Structure
- Priority Queue Introduction
- Deque (Introduction and Applications)
- Implementation of Deque using circular array

Lecture 15: Binary Tree.

Binary Tree:
- Binary Tree Introduction
- Binary Tree Properties
- Types of Binary Tree

Lecture 16: Binary Tree. (Continue)
- Handshaking Lemma and Interesting Tree Properties
- Enumeration of Binary Tree
- Applications of tree data structure
- Tree Traversals
- BFS vs DFS for Binary Tree

Lecture 17: Binary Tree. (Continue: 2nd)

- Level Order Tree Traversal
- Diameter of a Binary Tree
- Inorder Tree Traversal without Recursion
- Inorder Tree Traversal without recursion and without stack!
- Threaded Binary Tree
- Maximum Depth or Height of a Tree

Lecture 18: Heap.

Heap:
- Binary Heap
- Why is Binary Heap Preferred over BST for Priority Queue?
- Binomial Heap
- Fibonacci Heap
- Heap Sort
- K'th Largest Element in an array
- Sort an almost sorted array/

Lecture 19: Hashing.

Hashing:
- Hashing Introduction
- Separate Chaining for Collision Handling
- Open Addressing for Collision Handling
- Print a Binary Tree in Vertical Order
- Find whether an array is subset of another array

Lecture 20: Graph.

Graph:

- Introduction, DFS and BFS.
- Graph and its representations
- Breadth First Traversal for a Graph
- Depth First Traversal for a Graph
- Applications of Depth First Search

- Applications of Breadth First Traversal
- Detect Cycle in a Directed Graph
- Detect Cycle in a an Undirected Graph
- Detect cycle in an undirected graph

6 (ShivVerma)