C++

Course Outlines:

C++ Programs
- ✓ Key features of C++
- ✓ Identifiers and keywords
- ✓ Simple declarations, expressions and statements
- ✓ Basic I/O
- ✓ Layout
- ✓ Guidelines

Fundamental Data Types
- ✓ Built-in types
- ✓ Integer numbers
- ✓ Floating Point numbers
- ✓ Characters
- ✓ Booleans
- ✓ Assignment
- ✓ Compound Assignment
- ✓ Increment and Decrement
- ✓ Defining constants
- ✓ Type conversions

Composite Data Types
- ✓ Defining and using enumerations
- ✓ Built-in arrays and their limitations
- ✓ Using the vector class
- ✓ Built-in strings as character arrays
- ✓ Using the string class
- ✓ Defining and using structures

Control Flow
- ✓ Simple and compound statements
- ✓ Selection with if else and switch statements
- ✓ Conditional expressions
- ✓ Looping with while and for statements

Functions
- ✓ Declaring, calling and defining functions
- ✓ Overloading
- ✓ Default arguments
- ✓ Scope issues
- ✓ Pass by copy
- ✓ Pass by reference
- ✓ Inline functions
- ✓ Header files and source files
- ✓ Pitfalls and guidelines

Object Concepts
- ✓Object behaviour
- ✓Object state
- ✓Object identity
- ✓Object-oriented programming
- ✓Classes
- ✓Encapsulation

Using Classes
- ✓Associating functionality with data
- ✓Class definitions
- ✓Public and private
- ✓Queries functions and modifier functions
- ✓Struct vs class

Pointers
- ✓Concepts and syntax
- ✓Pointers to structured types
- ✓Pointers for encapsulated objects
- ✓Null pointers
- ✓Pointers vs. references

Implementing Classes
- ✓Defining member functions
- ✓Object identity
- ✓The this pointer
- ✓Initialisation
- ✓Constructors
- ✓Default constructors
- ✓Member Initialisation
- ✓Scope issues
- ✓Inlining member functions

Operator Functions
- ✓Operators as functions
- ✓global operators
- ✓member operators
- ✓I/O stream operators
- ✓Pitfalls and guidelines

Object Relationships
- ✓Associations and their implementation
- ✓Compositions and their implementation
- ✓Navigation
- ✓Delegation
- ✓Multiplicity

Dynamic Memory
- ✓The need for dynamic memory
- ✓Dynamic objects
- ✓Using new and delete

- ✓Dynamic arrays
- ✓Using new[] and delete[]
- ✓Destructors

More Pointers

- ✓Pointers and arrays
- ✓Pointer arithmetic
- ✓Pointers as array iterators
- ✓Pointers and const
- ✓Pointers vs. references

Containers

- ✓Container concepts and classification
- ✓Template classes
- ✓Standard containers
- ✓Vector
- ✓List
- ✓Iterators
- ✓Template functions
- ✓Algorithms

Copying

- ✓Copy construction
- ✓Copy assignment
- ✓Compiler generated copy behaviour
- ✓Problems
- ✓Solutions
- ✓Reducing Copying
- ✓Restricting Copying

Class Relationships

- ✓Extension of existing classes using inheritance
- ✓Polymorphic behaviour
- ✓Type substitutability
- ✓Abstract base classes

Inheritance

- ✓Protected members
- ✓Substitutability
- ✓Scoping
- ✓Base class initialisation
- ✓Order of object construction and destruction
- ✓Guidelines

Polymorphism

- ✓Declaring and defining virtual functions
- ✓Virtual destructors
- ✓Pure virtual functions
- ✓Using polymorphism through pointers and references
- ✓Guidelines