

## SYLLABUS

## COVERAGE

- Introduction to Programming
- Fundamentals in C++
- Control statements
- Pointer array Reference
- Function
- Introduction to oops
- Classes and Objects
- Constructors and Destructors
- Operator Overloading
- Inheritance and Composition
- Polymorphism
- Exception handling
- Input / Output in C++: Streams
- File handling
- Working with String
- Command line arguments
- Namespace
- Templates
- Data Structures(introduction)
- STL
- Process and Threads
- Graphics
- WEB development
- Project

## SYLLABUS IN DETAILS

## Introduction to Programming

- Program and Programming
- Programming Languages
- Types of software's Operating Systems
- Dos commands
- Basic Linux commands and vi editor
- Compiler, Interpreter, Loader and Linker

## Fundamentals in C++

- History of 'C++'
- Migrating from procedural oriented language
- to object oriented languages Program
- Keywords
- Variables
- Constants
- Data type
- Operators

- Manipulators and uses
- Basic Structure of a 'C++' program

## Control statements

- Conditional Control Statements
- if
- if-else
- nested if-else
- else-if ladder
- Multiple Branching Control Statement
- switch-case
- Loop Control Statements
- while
- do-while
- for
- Nested Loops
- Jump Control statements

- break
- continue
- goto
- exit
- return
- Programming Examples
- FAQ's

#### Pointer array Reference

- pointer variable
- Reference variable/alias variables?
- Reference to Reference variable?
- Reference to array?
- Reference vs normal variable?
- Reference vs pointer variable?
- 1D and 2D Arrays
- What is dynamic memory allocation?
- The new and delete operator
- new vs malloc
- delete vs free
- Dynamic 1D and 2D Arrays

#### Function

- What is function ?
- Why function ?
- Advantages of using functions
- Function Prototype
- Defining a function
- Calling a function
- Actual and Formal Arguments
- Types of functions
- Parameter Passing Techniques
- Call by Value
- Call by Reference
- Call by Pointer
- Return statement
- Returning More than one value From A Function
- Return by value mechanism
- Return by pointer mechanism
- Return by reference mechanism
- Inline Functions
- Default Arguments

- Function Overloading
- Lambda function.
- Recursion

#### Introduction to oops

- c structure vs c++ structure
- c++ class vs c++ structure
- Class
- Object
- Encapsulation
- Abstraction
- Polymorphism
- Inheritance
- Message Passing

#### Classes and Objects

- Declaring / defining classes
- Data members and member functions
- Access specifiers : public and private and protected
- Creating objects of a class
- Pointers to object
- Implicit this pointer
- Static data members
- Static member functions
- Passing objects to a member function
- Returning objects from a member function
- Friend functions
- Friend classes
- Nested classes
- Local classes
- The const member functions
- The const objects
- Array of objects
- static objects
- What are inline functions?

#### Constructors and Destructors

- Defining Constructor
- Defining Destructor
- Comparing Constructor Member Function
- Default Constructor

- Argument Constructor
- Copy Constructor
- Constructor Overloading
- Default Argument in Constructor
- Anonymous object
- Private Constructor and Destructor
- Local vs Global object

#### Operator Overloading

- Need of Overloading
- Defining Operator Overloaded Function
- Operator Overloading Rules
- Overloading Binary Operators
- Overloading Binary Operators using Friend
- Overloading Other Operators
- Overloading Unary Operators
- Overloading Unary Operators using Friend

#### Inheritance and Composition

- What is inheritance?
- The is-a relationship
- Single Level Inheritance
- Multilevel Inheritance
- Multiple Inheritance
- Name ambiguities under multiple inheritance
- Hierarchical Inheritance
- Hybrid Inheritance
- Multipath Inheritance
- Why virtual base classes?
- Constructor and Destructor in Inheritance.
- Common constructor.
- Delegation
- What is composition?
- The has-a relationship

#### Polymorphism

- About polymorphism
- Compile time and runtime polymorphism
- Virtual functions.
- Pure virtual function and abstract base class.
- What is RTTI (Run-time Type Information)?
- VPTR and VTABLE.

- Difference between member Function Overloading and Overriding
- Object slicing.
- Constructor and virtual function.
- Virtual destructor.
- Destructor with virtual function.

#### Exception handling

- What is an exception?
- Throwing an exception
- Catching an exception
- Trying for an exception
- Order of catch blocks
- Catching all exceptions
- Nested try blocks
- Rethrowing an exception
- Exception specifications
- What is stack unwinding?
- Exceptions in ctors and dtors
- The unexpected() function
- The terminate() function
- The standard exceptions
- Creating our own exception classes

#### File handling

- Hierarchy of File Streams
- Using constructor method
- Using open and close member function method.
- Object as file stream reader and writer
- Both sequential and random file accessing mechanism.
- Different error handling mechanism in files

#### Input / Output in C++: Streams

- Hierarchy of I/O Streams
- Fundamental stream classes and objects
- Standard input and output functions
- Formatting flags and manipulators

#### Working With String

- Different C string handling library
- string handling using relational operator
- Different string handling function

## Namespace

- Creating name space
- Using name space
- Nested namespace and anonymous namespace

## Command line arguments

- what is command prompt?
- why command line?
- What are command line arguments?
- Programs using command line

## Data Structures

- Introduction
- Single Linked List
- Circular Linked List
- Doubly Linked List
- Stacks
- Queues

## Templates

- What is generic programming?
- Need of Template
- What are function templates?
- Argument deduction and template parameters
- Overloading function templates
- What are class templates?
- Specializations of class templates

## STL

- STL Components
- Containers
- Iterators
- Algorithms
- Common container operations

- Vectors
- Deques
- Lists
- Sets and multisets
- Maps and multimaps
- Implementing reference semantics
- When to use which container?
- Special STL Containers
- Stacks
- Queues
- Priority Queues
- Bitsets
- STL Iterators
- Input iterators
- Output iterators
- Forward iterators
- Bidirectional iterators
- Random access iterators

## Database operation

- What is database?
- SQL for relational database.
- About API connect to database.
- Database connectivity MySQL.
- Database manipulation using C++
- Process and Threads
- Graphics
- WEB development
- Web basics.(HTML, java script, CSS).

## Project