**C++ Training Content**

**Basics**

* Introduction to C++
* Different paradigms of problem solving
* POP vs OOP
* Constants
* Variables
* Keywords
* Data types
* Declaration of Variables
* Output Stream (cout) & Manipulators
* Input Stream (cin)
* Comments
* Operators
  + Arithmetic operators
  + Relational operators
  + Logical operators
  + Assignment operators & compound assessment operations
  + Increment & decrement operators
  + Conditional operators
  + Bitwise operators
  + Shift operators
  + Type casting
  + Compound assignment operators
  + Address operators
  + Comma operator
  + Pointer operator
  + Sizeof operator
  + new operator
  + delete operator
* Control Statements
* Conditional Control Statements
  + If, if-else
  + nested if-else, if-else-if ladder
* Multiple Branching Control Structure
  + switch-case
* Loop Control statements
  + while
  + do-while
  + for
* Nested Loops
* Jump Control structures
* break
* continue
* goto
* return
* Arrays
* Strings
* Structures
* Pointers
* Dynamic memory allocation using new and delete

**Functions**

* Defining a Function
* Calling a Function
* Return statement
* Function Prototype
* Basic Function Designs
* Scope
* Reference variables
* Recursion
* Parameter Passing Methods
  + Call by value
  + Call by address
  + Call by reference
* Function Overloading
* Default Arguments
* Inline Functions

**Templates**

* Introduction
* Advantages
* Function Templates
* Over loading function template
* Class Templates
* Inheritance Class Templates

**STL (Standard Template Library )**

* Vector
* List
* deque
* Set
* Map
* multiset
* multimap
* Iterators and Algorithms

**Object Oriented Programming**

* Features of Object-Oriented Programming Languages
  + Object
  + Class
  + Abstraction
  + Encapsulation
  + Inheritance
  + Polymorphism
  + Dynamic Binding
  + Message Communication

**Classes and Objects**

* Defining a Class
* Creating Objects
* Access specifiers
* Accessing Class Members
* Scope Resolution Operator ( :: )
* Defining Member Functions
  + Outside the class
  + Inside the class
* Member function with argument
* This pointer
* Passing Objects as Arguments
* Returning Objects
* Array of objects
* Pointer to object
* Dynamic objects
* **Friend Functions**
* **Friend Class**
  + **Composition**
    - Container class
    - Contained class
  + Programs
  + Student Class
  + Employee Class
  + Complex Class
  + Matrix Class
  + Rational Class
  + Circle Class
  + Rectangle Class

**Constructors & Destructors**

* Constructors
* Properties of constructors
* Types of constructors
  + Default Constructors
  + Parameterized Constructors
  + Copy Constructors
* Constructor Overloading
* Constructors with Default Arguments
* Destructors
* Differences between Member functions & Constructors
* Differences between Constructors & Destructors
* Static Data Members
* Static member functions
* Constant data members
* Constant Member Functions

**Operator Overloading**

* Defining Operator Overloading Function
* Overloading Unary Operators
* Overloading Binary Operators
* Overloading Unary Operators using Friend Functions
* Overloading Binary Operators using Friend Functions
* Overloading << & >>

**Inheritance**

* Class hierarchies
* Base classes
* Derived Classes
* Derived Class Definition
* Access specifier : protected
* Types of Inheritance & Programs
  + Single inheritance
  + Multiple inheritance
  + Hierarchical inheritance
  + Multi-level inheritance
  + Hybrid inheritance
  + Multi-path inheritance
* Constructors in Derived Classes
* Destructors in Derived Classes

**Polymorphism and Virtual Functions**

* Static Binding
* Dynamic Binding
* Virtual Destructor
* Function Overriding
* Accessing Members using Pointers
* Virtual Functions
* Pure Virtual Functions
* Abstract Classes
* Virtual Destructors

**Exception Handling**

* Types of Errors
* Benefits of exception handling
* try, catch, throw keywords
* Throwing an exception
* ‘try’ block
* Catching an exception
* Exception objects
* Rethrowing an exception
* Exception Handling Mechanism
* Catching all exceptions
* Nested try blocks

**Files**

* File Streams Classes
* Opening & Closing a File
* Detection End of File
* File Pointers & Their Manipulation
* Sequential Files
* Random Access Files

**I-O Streams**

* I-O stream Class hierarchies
* Unformatted I-O Operation
  + get(), put(), getline()
  + write()
  + in cout
  + cin
* Formatted I-O Operations
  + width(), precision()
  + fill(), setf()
  + unsetf()
* Manipulators
  + Manipulator operators
    - Endl, ends
  + manipulator functions
    - setw(), setfill()
    - setprecision()
    - setiosflags()
    - setbase()
    - resetiosflags()
  + User defined manipulators
  + Operator and Overloading