

**Object Oriented Programming C++**  
**32Hours**

**Introduction:**

- What is object oriented programming?
- Why do we need object- oriented
- Programming characteristics of object-oriented languages
- C
- C++

**C++ Programming basics:**

- Output using cout
- Directives
- Input with cin
- Type bool
- The setw manipulator
- Type conversions

**Functions:**

- Returning values from functions
- Reference arguments
- Overloaded function
- Inline function
- Default arguments
- Returning by reference

**Object and Classes:**

- Making sense of core object concepts (Encapsulation, Abstraction, Polymorphism, Classes, Messages Association, Interfaces)
- Implementation of class in C++
- C++ Objects as physical object
- C++ object as data types constructor
- Object as function arguments
- The default copy constructor
- Returning object from function
- Structures and classes
- Classes objects and memory static class data
- Const and classes

**Arrays and string arrays fundamentals. Arrays as class Member Data:**

- Arrays of Object
- String
- The standard C++ String class

**Operator overloading:**

- Overloading unary operations
- Overloading binary operators
- Data conversion
- Pitfalls of operators overloading and conversion keywords
- Explicit and Mutable

**Inheritance:**

- Concept of inheritance
- Derived class and based class
- Class constructors
- Member function
- Inheritance in the distance class
- Class hierarchies
- Inheritance and graphics
- Public and private inheritance
- Aggregation : Classes within inheritance and program development

**Pointer:**

- Addresses and pointers
- The address of operator and pointer and arrays
- Pointer and Faction pointer and C-types string
- Memory management : New and Delete, pointers to objects, debugging pointers

**Virtual Function:**

- Virtual Function
- Friend function
- Static function
- Assignment and copy initialization
- This pointer
- Dynamic type information

**Streams and Files:**

- Streams classes
- Stream Errors
- Disk File I/O with streams
- File pointers
- Error handling in file I/O with member function
- Overloading the extraction and insertion operators
- Memory as a stream object
- Command line arguments
- Printer output

**Templates and Exceptions:**

- Function templates
- Class templates Exceptions

**The Standard Template Library:**

- Introduction algorithms
- Sequence containers
- Iteators
- Specialized iteators
- Associative containers
- Strong user-defined object
- Function objects