

# OOPS Programming with C++

## Synopsis & Objective of Course

This comprehensive training course is carefully designed to provide a complete mastery over C++ Programming. The trainees will initially understand the concepts of Object Orientation (OOPS) in the form of classes, objects, polymorphism, abstraction etc. And then delve into advanced concepts like, File handling, exception handling, templates, RTTI etc.

## Target Audience

Programmers & developers wanting to gain proficiency in C++ programming skills. Also for those who need to develop clarity on complex OOPS concepts.

## Prerequisites

Attendees should have a basic knowledge of programming and computers. A basic understanding of C Programming is essential.

## Delivery

The training will be instructor led, with each section of the material being covered by the trainer and followed by hands-on practical exercises. Training can be provided on GNU C++ compiler or any of the various windows based compilers.

Certain implementation oriented projects will be assigned to help trainees enhance their system programming skills.

**Duration** : 5 days

## Course Contents

Day	Topic	Sub topics
Day 1	Introduction to OOP	History of OOP, Structure of OOP, Disadvantages of OOP, Data types Variables Operators Conditional and Loop constructs Arrays Pointers Type casting Operators Bool data types structures in C ++ Functions.

	<b>Classes</b>	<p>Object oriented approach</p> <p>Difference between structures and classes,</p> <p>Constructor</p> <p>Destructor</p> <p>Types of constructor this pointer</p> <p>Functions within a class outside a class,</p> <p>Creating objects</p> <p>Object array</p> <p>Pointer object</p> <p>Memory allocation</p> <p>Access modifier</p>
<b>Day 2</b>	<b>Operator Overloading</b>	<p>Definition</p> <p>Need for it</p> <p>Why Operator Overloading</p> <p>Overloading arithmetic unary operator (example : complex, matrix classes, copy constructor, data conversion between objects of different classes)</p>
	<b>Inheritance</b>	<p>Concept</p> <p>Types of inheritance</p> <p>Over riding base class members,</p> <p>Access specifiers ,</p> <p>:: operators with over ridden members</p> <p>Base class initialization</p> <p>Multiple inheritance</p> <p>Problem and solution in multiple inheritance</p> <p>Inheritance using access specifiers</p>
<b>Day 3</b>	<b>Polymorphism</b>	<p>Late binding</p> <p>Early binding</p> <p>Types of polymorphism</p> <p>Virtual functions</p> <p>Friend functions</p> <p>Implementation of the above two functions</p> <p>Constructors &amp; Destructors in polymorphism</p> <p>Virtual base class</p> <p>Virtual constructors and virtual destructors</p>
<b>Day 4</b>	<b>Basic I/O And File concepts</b>	<p>Hierarchy</p> <p>Manipulators</p> <p>Istream, ostream and iostream classes</p> <p>Overloading &lt;&lt; &amp; &gt;&gt; operators</p> <p>File I/O</p> <p>Opening &amp; Closing of files</p>

<b>Day 5</b>	<b>Templates</b>	Generic programming Template class Function template Examples
	<b>Exception Handling and Advanced Features</b>	Classes within classes Friend class Namespaces RTTI Mutable & explicit keywords Need for it Try clause Throw an exception Catch block

### **Trainers' Profile**

Corporate Trainer(s) with more than 6 years of experience in embedded development & corporate training in CMM level5 companies.

### **Scheduled & On-site Training**

Apart from in-house training programs, comprehensive training can be also provided as per the requirement & will be optimally customized as per the client's needs.

*For training calendar, availability of seats & other details please mail us at [training@sigmasolutions.co.in](mailto:training@sigmasolutions.co.in)*