

C++ Programming

Duration- 4 Weeks

Perspective

1. The Software Crisis
2. Design Techniques
3. Large Software Systems
4. Roots of Object Technology
5. What Is Object-Oriented Programming?
6. C++ and Object-Oriented Programming
7. Why C++?
8. Features of C++
9. Pros and Cons of C++

The Language of Object-Orientation

1. What Is an Object?
2. What Is a Class?
3. Encapsulation
4. Data Hiding
5. The Public Interface
6. Relationships Among Classes
7. Inheritance
8. Polymorphism
9. Object-Oriented Design

C vs. C++

1. Comments
2. Namespaces
3. Simple Output
4. Simple Input
5. Definitions Near to First Use
6. Function Prototypes
7. The inline Specifier
8. const
9. Structure Members
10. The Reference Type
11. Overloading Function Names
12. Default Parameters
13. The Scope Resolution Operator
14. Aggregates
15. Operators new and delete
16. The bool Data Type
17. The string Data Type

Fundamentals of Classes

1. Data Types
2. User Defined Data Types
3. Using the Class Concept
4. Defining a Class
5. public and private Access Levels
6. The Scope Resolution Operator ::
7. Using Class Objects Like Built-in Types
8. Scope
9. Constructors
10. Member Initialization Lists
11. Destructors
12. Array of Objects
13. Pointers
14. The this Pointer
15. Passing Objects to Functions
16. Returning Objects From Functions
17. static Class Members

Operator Overloading

1. Introduction
2. Rules for Operator Overloading
3. Rationale for Operator Overloading
4. Overloading Member Functions
5. Overloading Non-Member Functions
6. friend Functions
7. The Copy Constructor
8. The Assignment Operator
9. Overloading []
10. Overloading Increment and Decrement Operators
11. const Objects and References

Composition of Classes

1. Relationships
2. Composition of Classes
3. The Point Class
4. The Line Class
5. Member Initialization Lists
6. An Application With Composition
7. The Copy Constructor Under Composition
8. operator= Under Composition

Inheritance

1. Introduction
2. Public Base Classes
3. The protected Access Level

4. Member Initialization Lists
5. What Isn't Inherited
6. Assignments Between Base and Derived Objects
7. Compile-Time vs. Run-Time Binding
8. virtual Functions
9. Polymorphism
10. virtual Destructors
11. Pure virtual Functions
12. Abstract Base Classes
13. An Extended Inheritance Example

I/O in C++

1. The iostream Library
2. Predefined Streams
3. Overloading operator<<
4. Overloading operator>>
5. Manipulators
6. Stream States
7. Formatted I/O
8. Disk Files
9. Reading and Writing Objects

Advanced Topics

1. Template Functions
2. Template Classes
3. Multiple Inheritance
4. User-Defined Conversions
5. Data Structures
6. An Iterator Class
7. Exceptions

Introduction to the Standard Template Library

1. Introduction
2. The Standard Template Library
3. Design Goals
4. STL Components
5. Iterators
6. Example: vector
7. Example: list
8. Example: set
9. Example: map
10. Example: find
11. Example: merge
12. Example: accumulate
13. Function Objects
14. Adaptors

Introduction

1. Background
2. Environmental Considerations
3. A Sample C Program
4. Variables and Data Types
5. Arrays
6. Example of a Program Using an int Array
7. Components of a C Program
8. C Operators
9. Examples of the Operators
10. Control Structures
11. Functions
12. Function Prototypes
13. Simple I/O

More I/O in C

1. The printf Function
2. The scanf Function
3. The Preprocessor
4. Conditional Compilation
5. Avoiding Multiple Inclusion for the Same File

Aggregates in C

1. Data Types Revisited
2. Aggregate Types
3. Arrays
4. Structures
5. Structures and Functions
6. Bit Fields
7. Enumeration Types

Pointers in C

1. Fundamental Concepts
2. Pointer Operations
3. Using Pointers to Alter a Function Argument
4. Using Pointers for Array Traversal
5. Pointer Arithmetic
6. Sending an Array to a Function
7. Command Line Arguments
8. Pointers vs. Arrays
9. Sending an Aggregate to a Function
10. Summary of the Uses of Pointers