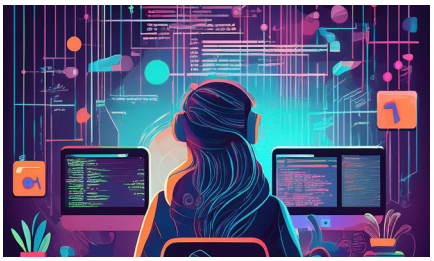
iMSD

C++ Course Core Components

INSTITUTE MEDIA STRATEGY DESIGN

tel 011 782 4297
fax 086 511 3343
Johannesburg, Cape Town,
Port Louis, London

msd@imsd.co.za
www.imsd.co.za



C++ Course Outline

Duration: 3 Days

Related Courses:
Phython, MATLAB, Java, C#, Lisp, Pascal, Scratch,

Course Overview and Objectives

This C++ course is designed to introduce students to the fundamental concepts and techniques of C++ programming. Whether you're a beginner or someone with some

programming experience in another language, this course will help you build a strong foundation in C++ and prepare you for more advanced topics in software development. The course covers essential programming principles, object-oriented programming (OOP), and modern C++ practices.

Pre-requisites:
Basic knowledge of mathematics (calculus and linear algebra) and programming (e.g., Python or C++) is recommended but not required.

Course Format:
Lectures, hands-on labs, assignments, and a final project.

Introduction to C++

- Introduction to the course
- Setting up the environment (C++ installation, IDE setup)
- Writing and executing the first C++ program

C++ Basics

- Syntax and structure of a C++ program
- Variables and data types
- Basic operators (arithmetic, comparison, logical)
- Input and output (cin, cout)
- Comments and documentation

Control structures

- Conditional Statements
- `if`, `else`, and `else if` statements
- Nested conditionals
- Switch statements

C++ loops

- `while` loop
- `for` loop
- `do-while` loop
- Nested loops
- Loop control statements (`break`, `continue`)

C++ functions

- Defining Functions
- Function declaration and definition
- Function arguments (by value, by reference)
- Return values
- Inline functions

Function overloading

- Concept and use cases
- Implementation of overloaded functions

Data structures

- Defining and accessing arrays
- Multidimensional arrays
- Array of strings

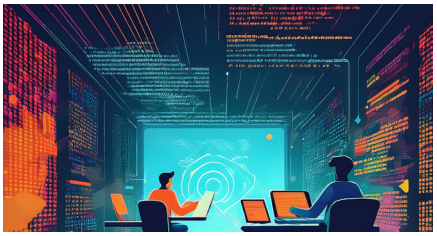
C++ Pointers

- Pointer basics
- Pointer arithmetic
- Pointers and arrays
- Pointers to pointers
- Pointers to functions

INSTITUTE MEDIA STRATEGY DESIGN

tel 011 782 4297
fax 086 511 3343
Johannesburg, Cape Town,
Port Louis, London

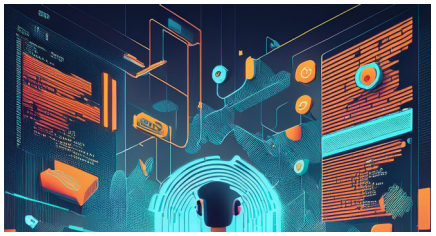
msd@imsd.co.za
www.imsd.co.za



Design and analyze robotic Systems:
Explain the components of C++ systems, including sensors, actuators, and controllers

Programming

Develop problem-solving skills:
Use functions, arrays, and pointers to create modular and efficient solutions to programming challenges.



Design process is key
Establish workflows for collecting, analyzing, storyboarding and producing visual stories.

C++ strings
- C-style strings
- String manipulation functions
- Standard string class (`std::string`)

Object Oriented Programming (OOP)
- Classes and Objects
- Defining classes
- Creating objects
- Access specifiers (public, private, protected)
- Constructors and destruc

Class Attributes and Methods
- Static variables and methods
- Constant member functions

C++ Inheritance
- Single and multiple inheritance
- Base and derived classes
- Accessing base class members
- Overriding member functions
- `virtual` functions and polymorphism

Operator overloading
- Concept and use cases
- Overloading arithmetic and comparison operators
- Overloading stream insertion and extraction operators

Encapsulation and Abstraction
- Encapsulation principles
- Abstract classes and pure virtual functions

Advanced topics
- Templates
- Function templates
- Class templates
- Template specialization

Exception handling
- Understanding exceptions
- `try`, `catch`, and `throw` keywords
- Custom exceptions

Standard template Library (STL)
- Overview of STL
- Containers (vector, list, deque, stack, queue, set, map)
- Iterators
- Algorithms (sort, find, reverse, etc.)

INSTITUTE MEDIA STRATEGY DESIGN

tel 011 782 4297
fax 086 511 3343
Johannesburg, Cape Town,
Port Louis, London

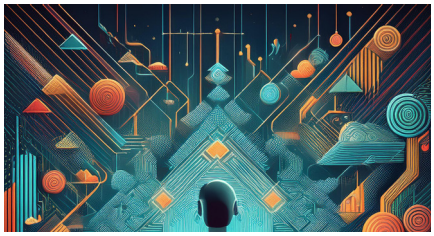
msd@imsd.co.za
www.imsd.co.za

Perform file operations

Read from and write to files, and handle file streams for data persistence.

Explore advanced C++ topics

Gain a basic understanding of templates, multithreading, and other advanced features for more complex applications.



Masterobject-oriented programming (OOP) process is key Implement and utilize classes, objects, inheritance, and polymorphism to build robust and reusable code.

File Handling

- Reading and writing files
- File streams (`ifstream`, `ofstream`, `fstream`)
- File modes
- Binary file operations

Dynamic Memory Management

- `new` and `delete` operators
- Dynamic arrays
- Smart pointers (`std::unique_ptr`, `std::shared_ptr`, `std::weak_ptr`)

C++ Multithreading

- Introduction to multithreading
- Creating and managing threads
- Thread synchronization (mutex, lock, condition variables)

Preprocessor Directives

- Macros and constants
- Conditional compilation
- File inclusion

Project Development

- Project Planning
- Choosing a project
- Project requirements and scope

Project Implementation

- Writing code
- Testing and debugging

Conclusion

- Summary of topics covered
- Best practices in C++ programming
- Further learning resources

We offer online support to clients on content covered on our courses.