



PGC-103: C++ Programming Introduction

Course Details

Course Code: PGC-103

Duration: 5 days

Notes:

- This course syllabus should be used to determine whether the course is appropriate for the students, based on their current skills and technical training needs.
- Course content, prices, and availability are subject to change without notice.
- Terms and Conditions apply

Elements of this syllabus are subject to change.

About this course

This course provides a practical hands-on introduction to the C++ programming language, current to the C++ 17 standard. Participants will learn the central concepts of the C++ language, with emphasis on the use of object-oriented techniques in writing robust code. Some advanced features are included.

At Course Completion

After completing this course, students will be able to:

- Create new classes
- Describe the process of data abstraction
- Create new data types using inheritance
- Use C++ class libraries
- Implement exception handling
- Write template functions and classes

Prerequisites

Participants must be fluent in ANSI C or a C++-like language such as Java or C#, and have an understanding of object-oriented concepts.

Academy IT Pty Ltd

Harmer House
Level 2, 5 Leigh Street
ADELAIDE 5000

Email: sales@academyit.com.au

Web: www.academyit.com.au

Phone: 08 7324 9800

Brian: 0400 112 083

Why C++?

- C vs. C++
- Abstract data types
- C++ as a better C
- I/O services
- Standard template library
- Standard compliance

Functions

- Function overloading
- Default parameter values
- Variable numbers of arguments
- Ambiguity
- Inline functions
- References
- The const and constexpr type-qualifiers

Classes

- Class definition
- Instantiating and using classes
- Member functions
- Classes and encapsulation
- Constructors and destructors
- Constructors and conversions

Storage Management

- Memory allocation
- Dynamic allocation
- Problems with dynamic allocation

Operator Overloading

- Simple operator overloading
- Return values of operator functions
- Predefined operator functions
- Conversions
- Member vs. non-member functions
- Overloading I/O operators
- Friend functions
- The effect of conversions
- Restrictions on overloading
- The [] operator

Initialization

- Initialization vs. assignment
- Constructors
- Arrays of objects
- Branching past initialization
- Constant class members
- Static class members

Inheritance

- Derivation

- Overriding names
- Constructor and destructor calls
- Inheritance at work
- Conversions
- Features not inherited
- Protected members
- When to use inheritance

Polymorphism

- Polymorphism
- Abstract classes
- New casting operators

I/O in C++ Programs

- Standard streams
- Insertion and extraction operators
- Manipulators
- Unformatted input and output
- File input and output
- Stream states
- Problems with object I/O

Exception Handling

- Exception handling concepts
- Exception handling terminology
- Uncaught exceptions
- Inheritance and exceptions
- When in an exception handler
- Function declarations

Templates

- Template functions
- Template classes
- Argument types
- Standard Template Library
- Implementation issues
- Vector of strings
- Range based for loop and auto
- Maps
- Lambda expressions
- Appendix: Review of C