incusdata (pty) ltd. 1995/04149/07
Coding · Matters
PO Box 148
Wierda Park, 0149
tel : +27 12 666 2020
info@incusdata.com
www.incusdata.com

Price: R15,800.00 excl. VAT
Duration: 5 days
Delivery: Virtual classroom or on-site training

Advanced C++ Programming

Description

This course will take you to the next level as a C++ programmer. You will learn advanced C++ programming concepts, and techniques to make your code more efficient and more portable. The course also covers new language features.

Objectives

After you have completed the Advanced C++ Programming course, you will be able to:

- Understand and use advanced C++ programming techniques.
- Understand and use the standard template library.
- Apply object-oriented techniques in C++ programs.

Intended Audience

You should attend the Advanced C++ Programming course if:

- You are a C++ programmer and you want to learn about the advanced aspects of the C++ language.
- You are a C++ programmer and you want to learn to write better C++ code.
- You are an engineer working with embedded systems written in C++.

Prerequisites

Before you attend the Advanced C++ Programming course:

- You must have attended our Standard C++ Programming course or already be comfortable with the fundamentals of the C++ programming language.
- You should have at least 6 months practical experience programming in C++.

Course Contents

Fundamentals

- Overview of C++ and ISO standards.
- C and C++ differences, portability issues and using C legacy code in C++.
- Structures and classes.
- Function overloading.
- Operators as functions.
- Default parameters.
- Variable number of parameters.
- Constants and macros.
- Inline functions.
- Namespaces syntax and usage.

- The standard library namespace.
- Practical issues and tips for using namespaces.
- Understanding and using references.
- Run time type information (RTTI).
- Name mangling.

Exception Handling

- Syntax, structure and usage.
- Exception handling options.
- Advanced error handling.
- Memory allocation error handling.
- Classes for exception handling.

Pointers and Memory Management

- Performance issues.
- Understanding new, delete and other memory allocation options in various environments.
- Overloading new and delete.
- Dangling pointers.
- Smart pointer classes.
- Reference counting.
- Shallow vs deep copying.

Classes, Inheritance and Polymorphism

- Inheritance concepts.
- Multiple inheritance.
- Virtual functions and the vtable.
- Abstract classes.
- Pure virtual functions.
- Access specifiers and usage.
- Private or protected constructors and/or destructors.
- Canonical classes.
- Iterator and envelope classes.
- Class factories.

Templates

- Overview.
- Template definition syntax.
- Function templates.
- Template parameters.
- Class templates.

- Deriving from templates.
- Smart pointers with templates.

Standard Template Libraries

- Principles and design issues.
- Practical use of the STL containers, algorithms, data models, adaptors and allocators.
- Tradeoffs.

Object-Oriented Design Principles

- Overview of Object Orientation and the Unified Modelling Language.
- Class relationships and associations.
- Containment, composiition and aggregation.
- HAS, USES and IS relationships.
- Interfaces and abstract classes.

** *The lecturer reserves the right to modify the contents of the course to suit the needs of the delegates.*