Advanced C++

Modality: Self-Paced Learning

Duration: 16 Hours

SATV Value:

CLC:

NATU:

SUBSCRIPTION: Learn, Master

About this course:

This course serves as the third course in a series of three courses designed for C++ developers and enthusiasts. Before attempting this course, it is highly suggested that participants complete the prior two courses in the Microsoft C++ learning series; namely "Introduction to C++" and "Intermediate C++". Nevertheless, this is not an absolute condition and a strong understanding of the basics of C++ make you eligible for this course. However, a clear understanding of memory allocation, pointers, file processing is absolutely necessary for understanding and excelling at this course. Participants should also have a good know -how of basic concepts of OOP.

The course outline includes an overview of the following concepts of C++ development; C++ templates, iterars , exceptions , advanced class mechanics and design patterns. The goal of this course is to train participants in multiple C++ software development techniques and skills.

Course Objectives:

Upon completion of the course, the participant should be able to have a sound knowledge and a superior skill set in the following ;

- Templates in C++
- C++ classes
- Patterns of design in C++
- C++ exceptions
- Object iterars
- Advanced mechanics of C++

Audience:

This particular course is aimed at the following audience;

- Developers of C/C++

Prerequisite:

The following prerequisites are absolutely necessary to be eligible to take this course;

- A good knowledge of pointers, memory allocation and file processing.
- Strong and sound basic OOP concepts

It is strongly recommended that participants complete both the “Introduction to C++” and “Intermediate C++” courses which serve as the first and second course in the Microsoft learning C++ series , respectively. However, this is not an absolute eligibility criteria.

Course Outline:

Exceptions

- Exceptions Recap
- Handling Exceptions
- Casting Exceptions
- Allocation
- Construction and Destruction
- Module 1 Review

Templates

- Template Functions
- Template Classes
- Template Adapters
- Module 2 Review

Iterars

- Iterar Introduction
- Standard Iterars
- Functions
- Defining Classes
- Module 3 Review

Patterns

- Function Pointers
- Interfaces
- Base Classes
- Inheritance
- The Singlen Pattern
- Namespaces
- Patterns
- Smart Pointers

- Module 4 Review