

Core Object-Oriented Design Patterns in C++ (TT1205)

Length: 3 days

Description

Throughout the course students will explore the following patterns, varying the levels of coverage to drill down on the most commonly used Patterns, and to simply survey others. Students will compare and contrast the patterns and explore the advantages and disadvantages of using certain patterns for explicit development functions the C++ environment.

Audience

This an introductory-level OO programming course, designed for developers who need to identify, design, and lead the implementation of OO projects. We will explore and apply the terminology, the specification, the processes and technologies specific to OO. Examples are written in Java.

Prerequisite

Attendees should be familiar with UML and have basic programming experience in C++. This course is not recommended for developers new to C++ programming.

Topics

SESSION: INTRODUCTION TO DESIGN PATTERNS

- Defining Design Patterns
- Why Use Design Patterns?
- Principles Behind Patterns
- Describing Patterns

SESSION: CREATIONAL PATTERNS

- Abstract Factory Design Pattern
- Singleton Design Factory
- Builder Design Pattern
- Factory Method Design Pattern
- Prototype Design Pattern
- Survey of Creational Patterns

SESSION: STRUCTURAL PATTERNS

- Overview of Structural Patterns
- Composite Pattern
- Adapter
- Proxy
- Bridge Pattern
- Façade Pattern

- Decorator
- Survey of Structural Patterns

SESSION: BEHAVIORAL PATTERNS

- Observer
- Strategy
- Iterator
- Visitor
- Interpreter
- Chain of Responsibility
- Command
- Mediator
- State
- Comparison and Summary

SESSION: FRAMEWORKS (OPTIONAL)

- Introduction to Frameworks

APPENDIX: INTRODUCTION TO UML (OPTIONAL)