

Object-Oriented Design Patterns & Frameworks in C++ (TT1255)

Length: 5 days

Description:

The course also contains several “thinking and drawing” lab exercises as a component of the object oriented overview portion of the training course.

This comprehensive training course will begin with a review of core concepts of Object Oriented analysis & design using UML (approximately one day). Throughout the remainder of the course we 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 in the C++ environment.

Audience

This an **introductory-level** programming course, designed for experienced C++ 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

Programmers with prior C++ and Object-Oriented exposure and background

Topics

SESSION: REVIEW OF OOAD USING UML

- Concepts of Analysis and Design
- Processes
- Activities
- Classes and Objects
- Providing Services
- Messaging
- Interfaces
- Relationships
- Types
- Dependencies
- Associations
- Generalizations
- Collaborations
- States and Activities
- Events
- State transition

SESSION: OBJECT ORIENTED ANALYSIS

- Use Cases
- Use Case Diagrams
- Defining Use Cases
- Extending Use Cases
- Variations On Use Cases

SESSION: OBJECT ORIENTED DESIGN

- Static Design Concepts
- Multiplicity
- Interface And Implementation
- Good Abstractions
- Constraints And Qualifiers
- Dynamic Design Concepts
- Sequence Diagrams
- Collaboration Diagrams
- Identifying Messages
- Characterizing Messages

Object-Oriented Design Patterns & Frameworks in C++ (TT1255)

- State Diagram
- Domain Design
- Domain Model
- Identifying Responsibilities
- Coupling

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: APPLICATION OF PATTERNS

- Patterns with Client Applications
- Patterns for the Data Tier
- Patterns in the Business Logic Tier
- Advanced Topics

SESSION: FRAMEWORKS (OPTIONAL)

- Introduction to Frameworks

APPENDIX: INTRODUCTION TO UML (OPTIONAL)