

Object Oriented Analysis and Design

Course Summary

Description

This course presents the key concepts and methodologies required to develop accurate programs.

Objectives

At the end of this course, students will be able to understand the key concept and methodologies required to develop accurate programs.

Topics

- Introducing OOAD
- Requirements and Initial Analysis
- Use Case Analysis
- Class Relationship Analysis
- Object Analysis
- Activity Analysis
- Class Design
- System Design
- From UML To Code and Back
- Introductions to Design Patterns and Frameworks

Audience

This course is designed for experienced programmers.

Prerequisites

This course is intended for both novice and experienced programmers who have a minimum of three months programming experience.

Duration

Three days

Object Oriented Analysis and Design

Course Outline

I. Introducing OOAD

- A. Defining OOAD Methodology
- B. The Analysis and Design Stage
- C. Defining the Analysis Phase
- D. Defining the Design Phase
- E. Characteristics of OOP
- F. Objects
- G. Classes
- H. Features of OOP
- I. Abstraction
- J. Encapsulation
- K. Association
- L. Aggregation
- M. Composition
- N. Inheritance
- O. Polymorphism
- P. Overview of Development Process
- Q. Lifecycle of Projects
- R. Major Phases of Lifecycle of Projects
- S. The Inception Phase
- T. The Elaboration Phase
- U. The Construction Phase
- V. The Transition Phase
- W. Workflow Within Phases
- X. Benefits

II. Requirements and Initial Analysis

- A. Purpose
- B. What is A Requirement?
- C. Gathering Information
- D. Avoid Traditional Assumptions
- E. Requirements Management
- F. Timing of Requirements Management
- G. Type of Requirement
- H. Gather Requirements Technique

III. Use Case Analysis

- A. Identifying System Boundaries
- B. Actors
- C. Finding Actors
- D. Use Cases
- E. Benefits of Use Cases
- F. Finding Use Cases
- G. Modeling Use Cases
- H. Creating Use Cases
- I. Scope

- J. Use Case Scenarios
- K. Primary Scenarios
- L. Secondary Scenarios
- M. Use Case Relationships

IV. Class Relationship Analysis

- A. Class Analysis
- B. Identifying Classes
- C. Entity Classes
- D. Control Class
- E. Boundary Class
- F. A Good and a Bad Class
- G. Finding Classes
- H. Noun/Verb Analysis
- I. Guidelines for Class Identification
- J. Golden Rules
- K. CRC Cards
- L. Responsibility
- M. Collaboration
- N. Types of Class Relationships
- O. Multiplicity
- P. Aggregation
- Q. Composition
- R. Association Class

V. Object Analysis

- A. Object State
- B. State Machine Diagrams
- C. Initial and Final States
- D. State Transitions
- E. State Action
- F. Self-Transitions
- G. Activity Analysis
- H. Activity Analysis
- I. Activity
- J. Action
- K. Control Flow
- L. Initial Node
- M. Final Node
- N. Object Flow
- O. Decision and Merge Node
- P. Fork and Join Node
- Q. Partition
- R. When to Use Activity Diagram?
- S. Interaction Diagrams
- T. Sequence Diagram

Due to the nature of this material, this document refers to numerous hardware and software products by their trade names. References to other companies and their products are for informational purposes only, and all trademarks are the properties of their respective companies. It is not the intent of ProTech Professional Technical Services, Inc. to use any of these names generically

Object Oriented Analysis and Design

Course Outline (cont'd)

- U. LifeLines
- V. LifeLine Start and End
- W. Messages
- X. Self Message
- Y. Lost and Found Messages
- Z. Communication Diagrams

VI. Class Design

- A. Visibility
- B. Inheritance
- C. Root And Leaf Classes
- D. Abstract Classes And Abstract Methods
- E. Interface
- F. Association Class
- G. Example

VII. System Design

- A. Packages
- B. Relationship of Packages
- C. Relationship of Packages – Example
- D. Components
- E. Component Diagram
- F. Deployment Diagram
- G. Node
- H. Node Instance
- I. Node Stereotype
- J. Artifacts
- K. Association

- L. Node as Container
- M. Example

VIII. From UML To Code and Back

- A. Adding Classes
- B. Adding Properties
- C. Adding Methods
- D. Generating Code
- E. Reverse Engineering
- F. Example

IX. Introductions to Design Patterns and Frameworks

- A. Introduction
- B. Patterns and Frameworks
- C. Patterns
- D. Frameworks
- E. How to Master software design
- F. Design Patterns
- G. Design Pattern Descriptions
- H. ClassLibraries vs. Frameworks vs. Patterns
- I. Comparing Pattern and Frameworks
- J. Types of Patterns
- K. Types of Design Pattern
- L. Design Patterns Overview
- M. Design Principles in Patterns
- N. Summary