DESCRIPTOR SYSTEMS

"We Bring You Up To Speed"

Object Oriented Analysis and Design with the UML

Description: This course teaches students the basic principles of object orientation and OO analysis and design. We will use the Unified Process and the Unified Modeling Language (UML) as tools. Illustrative programming examples use the Java language, but Java programming experience is not required.

Prerequisites: None

Audience: Business analysts, developers, managers and other people interested in object-oriented analysis and design.

Length: Three days.

Format: Lecture with hands-on lab exercises.

Objectives:

After taking this class, you will be able to:

1. Describe the three pillars of object-orientation and explain the benefits of each.
2. Create use case documents that capture requirements for a software system.
3. Create class diagrams that model both the domain model and design model of a software system.
4. Create interaction diagrams that model the dynamic aspects of a software system.
5. Explain the facets of the Unified Process approach to designing and building a software system.
6. Describe how design patterns facilitate development and list several of the most popular patterns.

Topics Covered:

Introduction to the Course

Why OOAD?

Problems and Issues with Traditional Techniques
The Rise of Object-Oriented Languages
Analysis versus Design

Modeling and Abstraction
Introduction to UML
Introduction to Java

Thinking in Objects

What is an Object?
What is a Class?
Creating Instances
The Three Pillars of Object Orientation
Object Oriented Programming Languages
Benefits of Object Orientation

Introduction to OO Analysis and Design

The Role of UML
The Importance of a Process
The Unified Process Overview
Using OOAD Tools

Introduction to the UML

UML History
UML Diagram Overview

Introduction to the Unified Process

Using Iterations
Benefits of Iteration
Timeboxing
The Amount of Ceremony
Using UML in the Process

Writing Use Cases

What is a Use Case?
Properties of a Good Use Case
Use Cases in an Iterative Process
What is an Actor?
Use Case Document Formats

Classes and Relationships

State and Behavior
Encapsulation
Navigation and Cardinality
Aggregation and Composition

Using Inheritance and Polymorphism

Has-A versus Is-a
Code Resuse
Polymorphism

Domain Analysis

What is Domain Analysis?
Domain Analysis in an Iterative Process
Starting with Use Cases
What are CRC Cards?

Creating the Domain Model Class Diagram

From Domain Model to Design Model

Using Interaction Diagrams
Using Design and GRASP Patterns
Using GRASP Patterns

Creating Interaction Diagrams

Sequence Diagrams
Collaboration Diagrams

Creating State Diagrams

Modeling Complex Objects
State Diagram Notation

Other UML Diagrams

Activity Diagrams
Package Diagrams

Design Model to Implementation

Writing Classes
Generating from a Tool
Reverse Engineering

Design Patterns

What are Design Patterns?
The Gang of Four Book