# Object-Oriented Analysis and Design

***CDT720***

This course provides an introduction to the process of analyzing the requirements for an application, and then designing an object-oriented approach to creating the application. The analysis phase includes discovering and documenting Use Cases. The design phase includes how to use diagrams from the Unified Modeling Language (UML) including Sequence Diagrams, Collaboration Diagrams, and Class Diagrams. Students will also be introduced to the concept of Design Patterns and will learn how to apply several of the most common Design Patterns. Throughout the course, students follow a single project from proposal stage through decisions about coding and testing. Students will create use cases and appropriate UML diagrams. They will not write code, but a fully coded example solution is available for study.

### Audience

- This class is intended for students who have some prior knowledge of object-oriented concepts and programming.
- The course does not require use of a particular programming language.

### Prerequisites

- Students must have some experience in an object-oriented programming language such as Java, C++, C#, or Visual Basic .Net.

### Course Length

- Three days

### Learning Objectives

- Use Cases
- Unified Modeling Language (UML)
- Domain Model
- Interaction Diagrams
- Class Diagrams
- Design Patterns
- Moving to Code

### Teaching Methods

- Lecture
- Written exercises
- Integrated case studies

## Course Outline

*DG5*

### Introduction

- Object-Oriented Analysis and Design
- The UML
- The Unified Process
- Iterative Development

### Use Cases

- Defining a Use Case
- Locating Use Cases
- Documenting a Use Case
- System Sequence Diagrams

### Domain Model

- Domain Analysis
- The Domain Model
- Concepts
- Associations
- Attributes
- Comments

### Interaction Diagrams

- Interaction Diagrams
- Sequence Diagrams
- Collaboration Diagrams

### Class Diagrams

- The Design Model
- Class Diagrams
- Associations
- Dependencies
- Inheritance

### Patterns

- Principles and Patterns
- Low Coupling / High Cohesion
- Facade
- Adapter
- Wrapper
- Factory
- Strategy
- Observer

### Additional Diagrams

- StateChart Diagrams
- Activity Diagrams
- Using Swimlanes
- Package Diagrams
- Object Diagrams
- Component Diagrams
- Deployment Diagrams

### Moving to Code

- Order of Coding
- Dependencies
- Testing
- Forward / Reverse Engineering