

# Object-Oriented Analysis and Design

605.704

1

## Course Description

- This course introduces the fundamental principles of object-oriented approaches to modeling software requirements and design.
- Topics include
  - strategies for identifying objects and classes of objects
  - specification of software requirements and design
  - design of class hierarchies
  - software reuse considerations
  - graphical notations
  - system implementation using object-oriented and object-based programming languages
  - comparison of object-oriented approaches to more traditional approaches based on functional decomposition.
- Case studies are used to illustrate and reinforce the underlying concepts.

2

## Learning Objectives

- By the end of this course you should be able to:
  - Apply an iterative process such as the Unified Process.
  - Analyze software requirements and document them using Use Cases.
  - Perform software analysis and record the results using UML notation.
  - Perform software design and record the results using UML notation.
  - Apply object-oriented patterns.
  - Discuss how object-oriented software development affects testing and quality.

3

## Prerequisites

- Foundations of Software Engineering (605.401) or equivalent background.
- Ability to program in an OO programming language (e.g., C++, Smalltalk, Java, C#, Python, Delphi, VB.net, etc.)

4

## Textbook

- The suggested textbook for the course is by Jim Arlow and Ila Neustadt, *UML2.0 and the Unified Process: Practical Object-Oriented Analysis and Design*.

5

## Project

- There is a project in this course.
- There are three deliverables required:
  - Requirements specification.
  - Analysis.
  - Design.
- You may either work by yourself, or on a team of 2 or 3 people on this project.

6

## Computer Requirement

- Access to a computer is recommended for the project, but not required. CASE tools are useful. You may use any tools at your disposal.

7

## Grading Criteria

- Test 1: 1/3
- Test 2: 1/3
- Project: 1/3

8