

# **95-706 Z3 -- Syllabus**

*2024, Spring Z3*

*Instructor: Constantine Aaron Cois, PhD*

## ***Description:***

Large-scale software development has been described as one of the most difficult of human undertakings. This course examines the reasons for the inherent complexity of software construction, and presents structured methods to deal effectively with it. The course will focus on the object-oriented approach for analysis and design. Students will gain an appreciation of the difference between writing programs and doing analysis and design. Problem formulation and decomposition (analysis) and solution building (design) will be covered. Students will work in small groups, each group having the responsibility for analysis, design and implementation of a software system. Case tools will be used in several stages of the development process.

## ***Learning Objectives:***

- Develop a working understanding of formal object-oriented analysis and design processes.
- Identify and describe the primary risks inherent to large-scale software development.
- Apply techniques and processes designed to mitigate these risks. Construct OOAD artifacts designed to enable these processes.
- Develop the skills to determine which processes and OOAD techniques should be applied to a given project.
- Prepare students for industry by developing an understanding of the application of OOAD practices from both software project management and software development perspectives.

## ***Prerequisites:***

Knowledge of an Object-Oriented language such as Java, Python, or C++ is required for this course.

## ***Office Hours:***

Office hours will be held weekly via Zoom meetings, on the date and time listed on the course calendar. The link to attend is:

<http://bit.ly/cois-office>

## ***Textbooks:***

Required Textbooks:

***Applying UML and Patterns, 3rd Edition, by Craig Larman***

Strongly Recommended Textbooks:

***Software Requirements*** Author: Karl Wieggers | Publication Date: August 25, 2013 | ISBN-13: 978-0-7356-7966-5 | Edition: 3rd ed.

***Design Patterns: Elements of Reusable Object-Oriented Software*** by Gamma, Helm, Johnson and Vlissides

***Grading:***

Grading is project-based. There will be 5 projects assigned, 3 of which will be team projects. Participating in projects and project teams is mandatory, and projects all build on the same case study as the term progresses. Missing time, even early in the term, will be severely detrimental to overall project performance, and thus course grade. If a student is unable to avoid joining the class late in the semester and has missed assignment(s), the weight of these assignments on their final grade will be evenly distributed over all other assignments.

***Assignment Submission:***

Assignments are due in digital form at the date/time shown on Canvas. Late submissions will receive a 10% grade reduction per day late.

***Policy on grading disputes:***

When questions arise in grading of a project, the following steps must be taken to address the issue:

1. If the student(s) disagree with a deduction, they will first contact the grader and ask for an explanation of the grade.
2. If the student(s) still disagree, they will write a clear, detailed email to the professor explaining why they feel points should not have been deducted.
3. The professor will make a final decision on whether points will be returned.

This procedure is mandatory to address grading issues.

***Lectures:***

In this distance section, we will refer to recorded lectures found here:

<https://heinzcollege.mediasite.com/Mediasite/Channel/mediasiteadmin-95-706-ooad-copyright-2016-carnegie-mellon-university/watch/c210655d0a114db4a509c514f02cf48e1d>

These recordings are from the last section I taught on campus, and all material is valid for this distance course.

Lecture material is supplemental to material covered in assigned readings, both textbook and case study. You are responsible for all information covered in lecture and supplemental videos, as well as information presented in posted slides and course readings.

### ***Grading Scale:***

The course grading scale is absolute (no curve). Grades are:

97.5-100	A+
92.5-97.4	A
90.0-92.4	A-
87.5-89.9	B+
82.5-87.4	B
80.0-82.4	B-
77.5-79.9	C+
72.5-77.4	C
70.0-72.4	C-

...and so on...

### ***Tentative Weekly Lecture/Reading Schedule:***

*Note: See Canvas site for exact schedule for this term, as some aspects may change*

#### Week 1

Topics:

- Introduction: What is OOAD?
- Software Development Life Cycle (SDLC) & Rational Unified Process (RUP)
- Requirements Elicitation & Analysis
- Quality Attributes
- Assignment of Project Teams

Reading for this week:

- Case Study: "Why Software Fails", IEEE 2005 (available on Canvas)
- Wiegers, Chapter 5, "Establishing the Business Requirements"
- Wiegers, Chapter 6, "Finding the Voice of the User"
- Wiegers, Chapter 8, "Understanding User Requirements"
- Wiegers, Chapter 10, "Documenting the Requirements"

#### Week 2

Topics:

- More on Requirements Analysis
- Case Study: POST
- Use Cases

Reading for this week:

- Wiegers, Chapter 14, "Beyond Functionality: Software Quality Attributes"
- Larman, Chapter 3, "Case Studies"
- Larman, Chapter 6, "Use Cases"
- Larman, Chapter 30, "Relating Use Cases"

### Week 3

Topics:

- From Use Cases to Functional Requirements
- Domain Model
- Associations
- Attributes
- Generalization

Reading for this week:

- Larman, Chapter 9, "Domain Models"
- Larman, Chapter 31, "Domain Model Refinement"

### Week 4

Topics:

- System Behavior
- System Sequence Diagrams
- Contracts
- State Diagrams
- Interaction Diagrams

Reading for this week:

- Larman, Chapter 10, "System Sequence Diagrams"
- Larman, Chapter 11, "Operation Contracts"
- Larman, Chapter 32, "More SSDs and Contracts"
- Larman, Chapter 29, "UML State Machine Diagrams and Modeling"
- Larman, Chapter 28 "UML Activity Diagrams and Modeling"

### Week 5

Topics:

- GRASP
- Determining Visibility
- Design Class Diagrams
- Law of Demeter

Reading for this week:

- Larman, Chapter 17, "GRASP: Designing Objects with Responsibility"
- Larman, Chapter 18, "Object Design Examples with GRASP"
- Larman, Chapter 25, "GRASP: More Patterns for Assigning Responsibility"

## Week 6

Topics:

- Class Design Principles
- Class Design Issues
- GoF Design Patterns

Reading for this week:

- Larman, Chapter 19, "Designing for Visibility"
- Larman, Chapter 16, "UML Class Diagrams"
- Larman, Chapter 26, "Applying GoF Design Patterns"
- Martin, "Design Principles and Design Patterns"

## Week 7

Topics:

- More GoF Design Patterns
- System Design/Architecture
- OO Metrics

Reading for this week:

- GoF, "Introduction to Design Patterns," (Chapter 1 of E. Gamma, R. Helm, R. Johnson, J. Vlissides, "Design Patterns", Addison-Wesley, 1994, pp. 1--31)
- Chidamber & Kemerer: "A Metrics Suite for O-O Design," (IEEE TSE '94)
- Basili et al. '95 (U. Maryland CS Dept. CS- TR-3443, at [www.cs.umd.edu/projects/SoftEng/ESEG/papers/CS-TR-3443.pdf](http://www.cs.umd.edu/projects/SoftEng/ESEG/papers/CS-TR-3443.pdf))