San José State University
College of Science/Computer Science Department
CS 286, Object-Oriented Analysis and Design, Section 3
Fall 2018

Course and Contact Information

Instructor:	Pearce
Office Location:	416 MacQuarrie Hall
Telephone:	(408) 924-5065
Email:	jon.pearce@sjsu.edu
Office Hours:	TR 13:30 – 15:00
Class Days/Time:	TR/15:00 – 16:15
Classroom:	MH 222
Prerequisites:	CS 160 or instructor consent

Course Description

This course focuses on the models built during the analysis and design phases of a typical software project. The analysis phase begins with the requirements model, where we learn how to identify actors, use cases, and scenarios. To understand our use cases we will need to understand the domain-specific concepts and relationships behind them. This happens when we build models of the application domain using UML class and activity diagrams. Warning: this often requires intense research into unfamiliar territory. Fortunately, we will learn a few analysis patterns that repeat themselves across many of these domains.

If the analysis models represent the user's view of a system, then the design model represents the developer's view. Constructed during the design phase, the fundamental unit of a design model is a collaboration. A collaboration consists of roles and interactions between objects playing those roles. We learn how to model collaborations using UML sequence diagrams. Common interactions are identified, and design patterns are introduced as recurring collaborations.

A good design can reduce future maintenance costs, but what is a good design? To answer this question we learn about principles, metrics, patterns, anti-patterns, and refactoring. Domain-driven design is presented as a collection of patterns for deriving the design model from the domain model by a process of distillation.

The hardest part of software design is deciding where to start. To answer this question we introduce several important architectural patterns such as the layered, pipeline, model-view-controller, client-server, agent-based, and component-container architectures. We implement these architectures as frameworks. This will often involve refining our architectures using some of the design patterns we

learned earlier. To prove the usefulness of our frameworks we build several simple applications on top of them.

Course Learning Outcomes

Upon successful completion of this course, students will be able to:

1. Elicit system requirements and express them in a requirements model.
2. Analyze domain concepts, relationships, and processes and express them in a domain model.
3. Create a design model from a domain model using design patterns.
4. Implement architectural patterns as frameworks in an OOP language.
5. Identify design flaws.
6. Create detailed UML diagrams using a CASE tool.

Required Texts/Readings

Textbook

There is no text for this course. Lecture notes will be posted at:

http://www.cs.sjsu.edu/faculty/pearce/modules/lectures/ooa2/index.htm
http://www.cs.sjsu.edu/faculty/pearce/modules/lectures/ood2/index.htm

Other Readings

My notes and lectures draw substantially from the following books:
Domain-Driven Design, Tackling Complexity in the Heart of Software; Eric Evans; Addison-Wesley; 2004.
Patterns, Principles, and Practices of Domain-Driven Design; Scott Millett with Nick Tune; Wrox; 2013.
Analysis Patterns: Reusable Object Models; Martin Fowler; Addison-Wesley; 1997.
Pattern-Oriented Software Architecture, volume 1; Buschmann, et. al.; Wiley; 1996.
Design Patterns, Elements of Reusable Object-Oriented Software; Gamma, et. al,; Addison-Wesley; 1994.

Required Software

Star UML 2; http://staruml.io/
Eclipse IDE for Java Developers; *http://www.eclipse.org/downloads/*

Classroom Protocol

Students are expected to bring their laptops to class.

Course Requirements and Assignments (subject to change with fair notice)

Assignments

There will be two types of assignments: labs and projects. Labs usually consist of multiple problems. These problems typically involve using a CASE tool to create UML diagrams. Projects involve creating an architectural model, implementing it as a framework, then implementing several customizations on top of the framework. (Coding is done in Java.) All assignments are submitted through Canvas. Rubrics will be used to grade the assignments. Assignments will typically be judged on accuracy, completeness, and implementability.

Examinations

There will be a midterm and a final exam. Both exams will be similar to the labs. Like the labs, the exams will be submitted through Canvas.

Course Schedule

The tentative course schedule below is subject to change.

Week	Topics/Activities	Assignment
1	Overview of software engineering. Requirements modeling.	
2	Requirements modeling.	1
3	Requirements modeling. Domain modeling	
4	Domain modeling.	2
5	Domain modeling. Analysis patterns.	3
6	Process modeling	4
7	Midterm review, Midterm	
8	Modeling collaborations	5, 6
9	Modeling collaborations. Design principles, patterns, and metrics.	
10	Pipeline architectures	7
11	Pipeline architectures	
12	Model-View-Controller architecture.	8
13	Model-View-Controller architecture.	
14	Agent-based architectures.	9
15	Agent-based architectures.	
16	Open Architectures, Final exam reviews on 12/6	
17	Study	
18	CS286 sec 3 final: M, 12/17, 14:45 - 17:00	

Grading Scheme

Course grades will be determined by computing a weighted average of all submitted work using the following weights:

Assignments	60%
Midterm	15%
Final	25%
TOTAL	100%

Weights of individual assignments appear on the course web page.

Assuming a normal distribution of weighted averages, I will use the following scheme for computing letter grades:

A	91% - 100%
A-	86% - 90%
B+	81% - 85%

B	71% - 80%
B-	66% - 70%
C+	61% - 65%
C	51% - 60%
C-	46% - 50%
D+	41% - 45%
D	31% - 40%
D-	26% - 30%
F	0% - 25%

University Policies

Per University Policy S16-9, university-wide policy information relevant to all courses, such as academic integrity, accommodations, etc. will be available on Office of Graduate and Undergraduate Programs' Syllabus Information web page at http://www.sjsu.edu/gup/syllabusinfo/