CS 356 Object-Oriented Design and Programming
Course Syllabus – Fall 2017

Instructor	Dr. Yu Sun
Email	yusun@cpp.edu
URL	http://yusun.io
Phone	909-869-3449
Office	8-13
Office Hours	Tuesday and Thursday 8am - 10am; or email me for an appointment
Course Session	Tu/Th 1:00pm - 2:50pm 8-345
Course URL	Blackboard

Course Focus

This course offers a survey and training of Object-Oriented (OO) design and programming principles, techniques and practices. In this course, you will be motivated to think and model problems in OO way, and solve problems using the OO principles, design patterns and practical OO libraries and tools. The main goal of the course is to teach you how to write more extensible, reusable, and maintainable code, and aid you in the ability to construct better software that meets industry standards.

Prerequisite

You should be comfortable writing code in Java. In addition, if you have not had the equivalent prerequisite of CS 241, you are advised to take it first.

Textbooks

All the course materials will be given in slides and available to download. No textbook is required. The following books are recommended to read that are helpful for some of the key topics related with this course.

[GoF95] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley, 1995.

[McLaughlin07] Brett McLaughlin, Gary Pollice, David West, Head First Object-Oriented Analysis and Design: A Brain Friendly Guide to OOA&D, O'Reilly Media, 2007.

[Martin08] Robert Martin, Clean Code: a Handbook of Agile Software Craftsmanship, Pearson Education, 2008.

Course Work and Grade Determination

There will be NO final exam for this course.

Programming Assignments
There will be 3 programming assignments that illustrate the value of good OO design and implementation techniques on the software lifecycle.

To encourage you to write well-designed software, we reserve the right to change the assignment specifications at any point before the due date. Expect this to happen at least once during the semester. If you have written your program in a modular fashion the changes will be trivial to implement.

The programs will be graded using the following criteria:
- 40% execution correctness
- 30% structure (e.g., modularization, information hiding, etc.)
- 10% insightful programming (e.g., developing reusable class components, etc.)
- 10% consistent style (e.g., capitalization, indenting, etc.)
- 10% appropriate commenting style

Mid-Term Exam

There will be one midterm exam (no final exam). The midterm will fully be based on material presented in class for the past weeks. It is therefore essential that you attend class in order to prepare for the exam. There will be no "makeup" midterm unless you ask permission from me before the exam.

Team Presentation

There will be one team assignment at the end of this course. Each team should choose one design pattern that is not covered in the lectures, and give a short presentation on it. The presentation should include: 1) the background and theories; 2) the running code example that applies the design pattern; 3) the example from an open-source project that applies this pattern. The details, schedules and requirements will be given around Week 6 or 7.

The relative weighting of each portion of the course is presented below:

• Programming Assignments (50%)
• Mid-Term Exam (30%)
• Team Presentation (20%)

Final grades will break at 90(A), 85 (B+), 80(B), 75 (C+), 70(C), 65, (D+), 60(D).

Honor Statement

Each student is to do his or her own work. This means that you are not to seek out the help of other students (or give help, if asked) in order to solve specific problems of your homework assignments. It also means that you should not sign up for mailing lists and ask for detailed help from others on the Internet. Of course, you may discuss generalities about an assignment with your fellow students. If you are unsure of what is permitted, in terms of discussing an assignment problem, please ask me for clarification.

Disabilities

If you have any disability that would put you at a disadvantage in performing an assignment, please meet with me privately to discuss ways in which I can assist you as you perform the required work in this course.

Tardiness

You are expected to arrive on time so that you do not cause a disruption in the middle of class. I would like to start the class at the scheduled time. If you cannot make it on time for some reason, please let me know. Persistent tardiness will be noted.

Tentative Schedule

This schedule and the order of the topics/assignments are subject to change.

Week	Day	Date	Description	Notes
1	Thu.	9/21	Course Introduction	
2	Tue.	9/26	Why Object-Oriented	
2	Thu.	9/28	Basic OOP in Java	
3	Tue.	10/3	OOP Principles in Java	A1 Assigned
3	Thu.	10/5	UML Basics	
4	Tue.	10/10	Version Control	
4	Thu.	10/12	OO SOLID Principles - 1	
5	Tue.	10/17	OO SOLID Principles - 2	

5	Thu.	10/19	Overview of Design Patterns	
6	Tue.	10/24	Design Patterns - 1	
6	Thu.	10/26	Design Patterns - 2	
7	Tue.	10/31	Design Patterns - 3	A2 Assigned
7	Thu.	11/2	Design Patterns - 4	
8	Tue.	11/7	Design Patterns - 5	
8	Thu.	11/9	Mid-Term Exam	
9	Tue.	11/14	Code Refactoring	
9	Thu.	11/16	Anti-Pattern	A3 Assigned
10	Tue.	11/21	Data Persistence	
10	Thu.	11/23	No Class - Thanksgiving Day	
11	Tue.	11/28	Team Presentation	
11	Thu.	11/31	Tech Job Interviews	