Fall 2015

SE 350: Object-Oriented Software Development

Instructor: Jane Cleland-Huang
Office: CDM 836, 312-362-8863
jhuang@cs.depaul.edu
Office Hours: Tuesday and Thursday – 3.30pm to 4.15pm

Course Objectives:

To expose students to principles and patterns of Object-Oriented Design and teach them to engineer elegant software systems. Students will become familiar with a wide variety of design patterns and Object-Oriented principles, will understand when each is relevant for use, integrate them into the design, and implement them as code. Students will develop greater proficiency in Java, Junit testing, and also use of a version control system. During this course students will design, develop, and test a non-trivial application, describe their work in a verbal presentation, and develop a professional portfolio of their work.

Class Schedule:

Tuesday and Thursday from 1.30pm-3.00pm.

Homework:

Homework will be assigned each week on Tuesday and will be due the following Monday night at midnight. All assignments must be submitted to subversion (instructions will be provided) Students may discuss solutions but all coding must be done individually. For this class – students are encouraged to use resources such as StackOverflow and may freely reuse code found at such sites. There will be no penalty for reuse – because this constitutes normal developer behavior. However, **all reused** code must be attributed. This means that if you reuse code you must provide a reference at the top of the class, or before the method, in which it is used. You may not reuse any code that is not available publicly (such as via Stack Overflow).

Late penalties will be strictly enforced at 10% per day. If you have a special need to turn in homework late you must either request permission one week in advance or provide evidence of the problem (doctor's note etc).

You should expect to spend an average of 5 hours working on programming each week. The more effort you put into this course, the more you will take-away, so I encourage you to go above-and-beyond wherever possible. If the assignment is easy for you and you finish in 2 hours – then try to add something to it. Extra credit may be awarded for significant additional effort. Any extra credit will be used to reduce the weight of the final exam. A maximum of 50% of the final exam grade may be earned by extra credit. You must still take the final exam – but the score will be condensed into a smaller portion of points.

SVN (SubVersion)

For this course we will be using SVN for version control. Each homework assignment may be pulled from your individual SVN repository. ALL references to files must be relative to the current homework directory and may not access external directories and/or files. Submit everything needed to run your program.

What if you get stuck?

This is a programming course and you may find some of the assignments challenging. There are several things you can do if you get stuck:

1. Avoid getting stuck in the first place. Attend lectures, read the chapters, and most importantly program and test iteratively. We will discuss this in class.
2. Come to office hours and ask for extra help. Don't come to office hours with a big ball of mud. Come with specific questions. ALWAYS be able to revert to the previous version of the code (SVN will let you do this). If you break something and you can't fix it – revert and start over. Commit frequently – every time you achieve a small success.
3. Start your program assignment early. Don't wait for the last 24 hours.
4. Post a question to the homework discussion forum on D2L. If you know the answer to somebody else's question – then feel free to post the answer. You can also earn additional points for being a good citizen and answering questions with correct and useful answers. If somebody answers your question and it solves your problem – please acknowledge this with a quick "Thank you".

Remember:

This is a course that can make a big difference in your programming maturity. Put lots of effort into it and walk away with new knowledge.

Reading List:

- (Required) Object-Oriented Design and Patterns, 2nd edition, Horstmann, John Wiley & Sons, 2005. ISBN: 978-0-471-74487-0
- (Optional) Head First Design Patterns, Freeman, O'Reilly Media, 2004. ISBN: 978-0596007126
- (Optional) Design Patterns, Gamma, Helm, Johnson & Vlissides, Addison-Wesley/Pearson, 1995. ISBN: 978-0201633610
- (Optional) Design Patterns Explained: A New Perspective on Object-Oriented Design, 2nd edition, Shalloway & Trott, Addison-Wesley/Pearson, 2005. ISBN: 978-0321247148

Grades:

Homework Assignments (Weeks 1-6)	25%
Final Project (Weeks 7-10)	25%
MidTerm	25%
Final	25%

Schedule

See next page

	Tuesday		**Thursday**		Reading
Sept.	8th		10th	L1. Course Introduction Basic UML (Class diagrams and Sequence Diagrams)	Review Horstman Chapter 1
	15th	**Lab: The Eclipse Environment + Programming Challenges**	17th	Our first design pattern: Strategy	Horstman Chapter 2, Chapter 3 section 3.1-3.4
	22nd	Programming by Contract	24th	Polymorphism and Inheritance	Horstman Chapter 3.5 to end, Chapter 4 Sec 4.1-4.5
	29th	**Lab: Introduction to GUI Prog**	1st	Unit testing	Horstman Chapter 4.6 to end
Oct.	6th	Design Patterns	8th	Design Patterns (continued)	Readings to be assigned as order of design patterns is determined.
	13th	**Lab: Midterm Practice Challenges**	15th	**Midterm exam**	
	20th	The open-closed principle	22nd	Design Patterns (continued)	
	27th	**Lab**	29th	Multi-threading	
Nov.	3rd	Java Object Model	5th	Java Object Model	
	10th	**Lab**	12th	Design Patterns (continued)	
	17th	Currently open…	19th	**Game Jam**	
	24th	**Final**			
All lab sessions will be held in CDM 658					