Syllabus

General Info

Course CS321 Software Development Practicum

Instructors
- Charilaos Skiadas (skiadas at hanover dot edu)
- Theresa Wilson (wilsont at hanover dot edu)

Term Spring 2018-2019

Offices LYN 102

Office Hours By appointment

Book
- *Agile Software Development, Principles, Patterns, and Practices*, by Robert Martin
- online ACM resources[1]

Websites
- Notes etc[2]
- Moodle[3]

Class times MTWRF 10:15am-5:00pm, in LYN120A, with break for lunch

Course Description

Software Development is a complicated process. Of course you have to write code that solves a problem, but this is only a small part of the work you have to do. You will invariably have to revisit code you have written a long time ago in order to improve it or to fix bugs. You will need to maintain "development" versions of your code along with "deployment" versions. You need ways to keep track of bugs in your code and proposed enhancements, and your progress towards them. And above all, you need to work with other people's code and they need to work with yours.

This course is a survey and practicum on all of these aspects of software development. It moves on parallel across 5 categories of topics:

1. *Java and Object-Oriented-Programming Fundamentals* reviews and reinforces the key aspects of OOP with a discussion of classes, inheritance and delegation, interfaces, encapsulation etc.
2. *Design Concepts and Principles* discusses the SOLID principles whos intent is to keep large code bases clean.
3. *Design Patterns* discusses tried-and-true solutions to common design problems.

[1] http://learning.acm.org
[2] https://skiadas.github.io/SoftwareDevelopmentPracticumCourse/
[3] https://moodle.hanover.edu/course/view.php?id=825

4. *Development Practices* covers standard development techniques such as agile methods, user stories and CRC Cards, version control fundamentals, test-driven development, issue management, and UML diagrams.
5. *Coding Practices* discusses topics related to the particulars of writing clean and readable code, focusing on general values and principles as well as more specific topics such as naming of variables and methods, commenting and formatting principles, the law of Demeter, exception handling, and refactoring techniques.

Along the way, students are also engaged in a collaborative project that will allow them to reinforce all these concepts while working with their peers on a large-scale project.

Textbook

There are two main sources of material for the course. The first is the main course textbook, Agile Software Development, Principles, Patterns, and Practices, by Robert Martin. This covers *Design Concepts and Principles* as well as *Design Patterns*.

The second is a list of electronic resources available through the ACM (Association for Computing Machinery) membership, which I am asking students to get. The ACM delivers resources that advance computing as a science and a profession, and this includes the ACM Learning Center[4] which provides you access to hundreds of online books and videos related to computing. The annual student membership fee for the ACM is around $20, a fairly modest amount given the resources it provides. I encourage you to continue your membership every year and to take advantage of the opportunities and resources it offers.

A catered list of reading material from electronic resources will be provided, and it will cover the remaining topics of *Java programming and OOP Fundamentals*, *Development Practices* and *Coding Practices*. Sections from the following books will be covered:

- *Clean Code*, by R. Martin
- *Clean Code* Video Series, by R. Martin
- *Design Patterns* Video Series, by R. Martin
- *Test-driven development, by example*, by K. Beck
- *Implementation Patterns*, by K. Beck

Course Components

Reading Assignments

In the class schedule page[5] for each class day you will find a list of links "prep" items. These may be videos to watch or book sections to read, possibly with a short assignment

[4] https://learning.acm.org/
[5] skiadas.github.io/SoftwareDevelopmentPracticumCourse/site/schedule.html

to help you learn the material. You are expected to do this prep work before coming to the next class.

Class Participation

You are expected to attend every class meeting. The accelerated nature of Spring term and the collaborative nature of the programming project require constant engagement with the material and your classmates. The class is typically divided in 5-6 *activities*. Missing an activity will result in a reduction of your final score by one percentage point, up to a total of 10 points. Emergencies will be dealt with on an individual basis. There are very few reasons that would qualify as an excuse for an absence. If you have known conflicts discuss them with us by the end of the first day.

Exams

There will be three midterms, on Monday 5/6, Monday 5/13 and Wednesday 5/22.

Project

For a large part of the course you will be engaged in a collaborative project with a group of your classmates. Project work will typically be as follows:

- We will be working on 2-3 day "development iterations" during which the group will be working on a set of features for the project.
- Each feature is "owned" by one student, who sees it through to its completion.
- At any given time, you will be either developing the feature you own, or you will be pairing up with another student to help them on their feature.
- Your grade is not determined by the "number of features you have completed" (beyond a minimum requirement). A large part of the grade is based on the success of the project as a whole. So assisting others with their features is just as important as working on your own features. Paired programming is both required and expected.

The project will overall count for 45% of the final grade, broken down as follows:

- Group Assessment
  - Evaluation of the overall quality of the project's codebase, based on a development rubric. (20%)
  - Assessement of the project success based on the delivered functionality. (10%)
- Individual Assessment
  - Leading development of a required number of features. (5%)
  - Effective paired programming with teammates on their features. (5%)
  - Peer assessment of your contribution to the team. (5%)

Phone policy

Your phone is required to be in your bag, set to silent, at all times, except during scheduled breaks (there will be at least one 5-minute break every hour). This includes both lecture and other in-class activities as well as programming/development time. If phone use becomes a problem, we will confiscate your phone until the end of the class day.

Grading

Your final grade depends on class participation, work on project, and three exams, as follows:

Components	Percent
Participation	10%
Project (Group)	30%
Project (Individual)	15%
Exam 1	15%
Exam 2	15%
Exam 3	15%

This gives a number up to 100, which is then converted to a letter grade based roughly on the following correspondence:

Letter grade	Percentage Range
A, A-	90%-100%
B+, B, B-	80%-90%
C+, C, C-	70%-80%
D+, D, D-	60%-70%
F	0%-60%