Syllabus

General Info

Course CS321 Data Wrangling and Management and CS328 Data Management and Web Services

Instructor Charilaos Skiadas (skiadas at hanover dot edu)

Term Spring 2018-2019

Office LYN 108

Office Hours By appointment

Book

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

Websites for notes[2]

Class times MTWRF 10:15am-12:30pm, 2:40pm-5:00pm, in LYN120A

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/

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[3] 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:

- *Java, a beginner's guide*, by H. Schildt
- *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

[3] https://learning.acm.org/

Course Components

Reading Assignments

In the class schedule page[4] you will find, for each class day, a list of links to reading assignments. Your homework will require you to have a solid understanding of the material covered there, so I strongly encourage you not to get behind.

Class Participation

You are expected to attend every class meeting, morning and afternoon. The accelerated nature of Spring term and the collaborative nature of the programming project require constant engagement with the material. You are only allowed to miss 1 period without excuse. From that point on, every missed period will result in a reduction of your final score by one percentage point, up to a total of 5 points. Excused absences should be arranged in advance, and backed by appropriate documentation. Emergencies will be dealt with on an individual basis. There are very few reasons that would qualify as an excuse for an absence.

Exams

There will be frequent quizzes based on the reading, as well as three midterms, on Friday 5/3, Friday 5/10 and Friday 5/17. In terms of your final grade, the exams you did better on will weigh more.

Project

For a large part of the course you will be engaged in a collaborative project with a group of your classmates.

Getting Help

- You should never hesitate to ask me questions. I will never think any less of anyone for asking a question. Stop by my office hours or just email me your question, which has the great benefit of forcing you to write it down in clear terms, which often helps you understand it better.
- You are allowed, and in fact encouraged, to work together and help each other regarding the class material, as well as the topics related to the lab assignments. However, you may NOT directly help each other with your lab assignments.

[4] skiadas.github.io/DataWranglingCourse/site/schedule.html

Grading

Your final grade depends on class attendance, homework, midterms and the final, as follows:

Component	Percent
Participation	5%
Quizzes	15%
Project	35%
Worst Exam	10%
Middle Exam	15%
Best Exam	20%

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%