Syllabus
COMP 335
Software Engineering

Fall 2019

1 Logistics

- **Where:** Center for Science and Business (CSB), Room 303

- **When:** MTWF 9–9:50am

- **Instructor:** Logan Mayfield

  - *Office:* Center for Science and Business (CSB), Room 344
  - *Phone:* 309-457-2200
  - *Website:* `http://jlmayfield.github.io/`
  - *Email:* lmayfield *at* monmouthcollege *dot* edu
  - *Office Hours:* Tu 1-2pm, W 1:30-3pm, Th 2-3pm, F 10-11:30am, or by appointment.

- **Website:** `http://jlmayfield.github.io/teaching/COMP335/`

- **Credits:** 1 course credit

Note: This Syllabus is subject to change based on specific class needs. Significant deviations from the syllabus will be discussed in class.

2 Description, Content, and Learning Goals

COMP335 introduces students to the basic principles and best practices of software engineering. Special attention will be paid to Agile programming methods and principles as well as practices for building software as part of a team of developers. Students taking this course will gain knowledge in: the basics of managing a software development project, how to gather program requirements, the principles for writing, debugging, and maintaining clean code in a team setting, the ethical obligations of software engineers, and the fundamentals of software architecture and design.

In this course students will combine reading and study with practice. Using the textbook as a starting point, students will work as a team to develop notes and resources suitable for supporting a larger scale software development project. As the semester progresses, the class will begin the development of such a project. Work will be carried out using open-source development practices and the suite of supporting tools found on GitHub. Projects will vary from semester to semester but the end goal will always be to release a working piece of software. Emphasis will be placed on participating in the software development process rather than traditional homework and exams.

2.1 Textbook

Dooley, John F. *Software Development, Design, and Coding.* Second Edition. Apress. 2017. ISBN:978-1-4842-3152-4

2.2 Software

When the emphasis of the course is on reading and study, the question and answer software *Socrative* will be used on an almost daily basis. Students will need to create a free student account at `https://socrative.com` where it is also possible to participate in socrative sessions via the web. Socrative also provides free iOS and Android apps for students to use. You can find them on the respective app stores or get direct links at `https://socrative.com/apps/`.

3 Workload

The course workload is as follows:

Category	Number of Assignments
Chapter Reading Notes	10-12
Peer Code Reviews	4-6
Quizzes	8-10
Software Development Project	1
Final Self-Reflection & Self-Evaluation	1

Participation

It is essential that every student attend class and come to class prepared. Some times this means having done the required required reading and being ready to respond to and discuss questions on socrative. Other times this means having made contributions to one more code or issue-tracking contributions to a GitHub repository and being ready to engage teammates in face-to-face discussion about those contributions. As the work on the development project progresses, the class will carry out peer code reviews. These too are an essential part of participating in the course and students will be evaluated on the quality of feedback they provide to others during these reviews.

Reading Notes

The class will work together to develop a set of outlines, summaries, references, and review questions to accompany most of the chapters in the textbook. These will be developed as Markdown pages in a basic GitHub repository and treated as if they were a piece of software. Every student is expected to make regular, constructive, and timely contributions to this effort. This will be tracked through GitHub's project tracking and management features. Students will also have the opportunity to make a case for their personal contributions if they feel these features do not adequately represent their contributions.

Quizzes & Final

There will be quizzes over the material in the textbook. They are meant to ensure that each individual student has a solid grasp on the basic principles of software engineering. The quizzes are not heavily weighted in the overall course grade and will often be composed of socrative style questions and possibly review questions written by the class itself as part of the reading notes development. The final exam is not an exam but a self-evaluation and reflection on the class project and the ways in which the principles of software engineering presented in the book manifested themselves in the project. In effect, it is practice for software engineering job interviews where an applicant might be asked specifics about their prior experience developing software. Students will also use that time to highlight their individual contributions to the project and provide context to the contribution metrics tracked on GitHub.

Project

The main aim of this course is to learn software engineering by engineering software. After some preliminary study, the class will begin work on a larger scale software development effort in which every member of the class works towards the same goal. The instructor will, for the most part, play the part of the client and

not a member of the development team. The project will be carried out using the Agile methods discussed in the text and open-source practices facilitated by GitHub. A finished product or prototype is expected by the end of the semester.

3.1 Course Engagement Expectations

The weekly workload for this course will vary by student but on average should be about 13 hours per week. The follow tables provides a rough estimate of the distribution of this time over different course components.

Assignment Type	Time/week
Scheduled Meetings	4 hours/week
Reading & Study	1-2 hours/week
Project Work (outside of class)	5-7 hours/week
	10-13 hours/week

4 Grades

This course uses a standard grading scale where percentage grades translate to letter grades as follows:

Score	Grade
94–100	A
90–93	A-
88–89	B+
82–87	B
80–81	B-
78–79	C+
72–77	C
70–71	C-
68–69	D+
62–67	D
60–61	D-
0–59	F

Students are always welcome to challenge a grade that they feel is unfair or calculated incorrectly. Mistakes made in the student's favor will never be corrected to lower a grade. Mistakes not in the student's favor favor will be corrected. *Basically, after the initial grading, a score can only go up as the result of a challenge.*

4.1 Grade Weights

The final grade is based on a weighted average of particular assignment categories. Students should be able to estimate your current grade based on your scores and these weights, but you may always visit the instructor *outside of class time* to discuss your current standing and check on some or all of the current course grade.

Category	Weight
Participation - Reading	10%
Reading Notes Contributions	10%
Participation - Peer Reviews	10%
Quizzes	10%
Final Self-Reflection & Evaluation	10%
Project Contributions	50%

4.2 Participation, Attendance, & Late Assignments

This course will make almost daily use of Socrative for in-class question and answer sessions. Questions will cover portions of the text that were assigned as reading and will range from simple checks to see if the reading was done to more challenging questions that follow from a close examination of the reading. For the most part, the only requirement is to provide an answer to every question and participate in the resultant discussions. On occasion, questions will be evaluated for their correctness and performance on these questions will also factor into the course participation grade. Students who do the reading and start the homework as soon as possible will have very little to worry about.

While there is no strict attendance policy, the course participation grade is based in large part on engagement with socrative. Absent students cannot participate in socrative sessions. Students should avoid unexcused absences, as defined in the college-wide absence policy. Whenever possible, let the instructor know of the absence before it occurs. When unexcused absences do occur, it is the student's responsibility to make up for the lost class time and to seek the permission of the instructor to hand-in or complete assignments that are late due to an unexcused absence.

In general, assignments are due at the specified time and no late assignments will be accepted unless an extension was requested prior to the due date. There are, of course, exceptions to this rule and students needing extra time can always contact the instructor for an extension. Do not just give up and eat a zero for the assignment. Ever. There is no penalty in asking for an extension nor is there a limit on extensions. That being said, there is no guarantee an extension will be given without legitimate need.

This course is designed around the assumption that students *engage in new ideas before they're covered in class meetings*. This means doing assigned reading, taking a stab at homework problems, and as a result coming to class and lab with some understand about a new idea or, just as likely, with a host of questions about something encountered in the reading and homework. Not attending class, skipping lab, and putting off work to the point that an extension is needed are signs that a student isn't holding up their end of the bargain and is not prepared to participate in class.

4.3 Calendar

This calendar is subject to change based on the circumstances of the course. A detailed, day-by-day calendar of reading requirements and expected quiz dates can be found on the course website.

Week	Dates	Chapter(s) Covered
1	8/21–8/23	1
2	8/26–8/30	2,3
3	9/2–9/6	3,4,18
4	9/9–9/13	18
5	9/16–9/20	5,6
6	9/23–9/27	
7	9/30–10/4	14
8	10/7–10/11	FALL BREAK (W-F)
9	10/14–10/18	15,16
10	10/21–10/25	17
11	10/28–11/1	7–11
12	11/4–11/8	7–11
13	11/11–11/15	7–11
14	11/18–11/22	7–11
15	11/25–11/29	THANKSGIVING (W-F) 7–11
16	12/2–12/7	READING DAY (Th). Final 12/7 at 8am