

# **COSC 340: Software Engineering, Fall 2021**

University of Tennessee, Knoxville

Dr. Austin Henley

**Room:** Min Kao 622

**Time:** Tue/Thu 2:50pm-4:05pm

**Textbook:** No textbook

**Website:** <https://tiny.utk.edu/cosc340>

**Instructor:** Dr. Austin Henley

**Office:** 353 Min Kao

**Office Hours:** By appointment

**Email:** [azh@utk.edu](mailto:azh@utk.edu)

**Teaching Assistants:** Dylan Lee, Ben Klein

**Office Hours:** By appointment

**Email:** [dlee97@vols.utk.edu](mailto:dlee97@vols.utk.edu), [bklein3@vols.utk.edu](mailto:bklein3@vols.utk.edu)

## **Course Goals**

The course aims to prepare students for real-world software engineering tasks. This will include learning widely-used tools (e.g., git and VS Code) and working in a group to build a large software application that students can include in their portfolio/resume. Additionally, students will learn and apply processes that software engineering teams utilize to manage software projects (e.g., agile development and code reviewing).

## **Topics**

- Software Requirements
- Software Analysis & Design
- Software Testing
- Software Configuration Management
- Software Project Management
- Software Engineering Process
- Software Engineering Research

## **Skills & Tools**

- Version control (git and GitHub)
- Integrated development environments (VS Code)
- Graphical user interface design and implementation
- Debugging
- Program analyzer

- Unit testing
- Documentation generator
- Python
- Code reviewing
- Usability testing
- If time permits: interview problem solving and contributing to an open-source project

## **Lectures and In-Class Activities**

Most class time will be used for lectures. The slides will be made available on the course website. There will occasionally be in-class activities (e.g., writing a user story, meet with your group, short quiz, etc.). Although attendance will often not be explicitly recorded, missing a class with an activity, whether it is announced or unannounced, will result in a zero unless a documented excuse is provided or prior approval from the instructor. There may also be live demonstrations on how to use specific software for the group project. These demonstrations may not be recorded or have any notes associated with them, so you are expected to attend and take any notes that you may need.

## **Assigned Reading**

To supplement the lectures, I will occasionally assign reading that should be done outside of class. There is no required textbook, so I will provide any document or link to the content (I might also assign YouTube videos!). I aim for these to be helpful and enjoyable to read and will avoid anything that requires considerable memorization.

## **Quizzes**

There will be frequent quizzes that cover the lectures, live demos, and assigned readings (approximately one per week). The quizzes will be posted on Canvas, though I may require that you take them in class. You can **not** make-up any quizzes. These are to be done individually and without any other resources (i.e., no friends, websites, books, etc.).

## **Group Project**

The focus of this course is a group project that consists of designing, building, and evaluating a software application. Students will work in groups of 3-5 to implement a graphical desktop application while using specific development tools and processes to meet milestones. Over a series of 2-3 week iterations, the group will iteratively implement the software while applying a variety of software engineering techniques (e.g., version control management, user stories, unit tests, code reviewing, etc.). The group will be periodically given portions of class time to meet. Groups are expected to communicate regularly outside of class and setup their own system and schedule for doing so (e.g., every Tuesday evening on Slack). Grades will be given for each iteration to the individual based on their contributions (planning, coding, etc.), tracked via GitHub commits and GitHub's issue feature. Group grades will be given based on the software's

functionality and on in-class activities throughout the semester. I may assign specific tasks to be done during specific iterations. Additionally, individual contributions may be verified at any point to ensure features are implemented as claimed. (I may re-grade individual contributions as necessary.)

The expectation is that contributions will be made frequently and regularly. **Iterations are not deadlines.** Committing your contributions only at the end of the iteration will result in a failing grade.

Although I will make the requirements and expectations as clear as I can, grading productivity and contribution quality is inherently **subjective**. Software development projects are often fuzzy with ever-changing requirements. Have fun with it!

Note about using someone else's code: It is common while programming to use source code found from other resources (e.g., examples, tutorials, Stack Overflow). This is completely acceptable **if** you provide credit. For example, if you use a function from an online resource, add a comment with the URL of the resource. Even if you make considerable changes to it, it is best to add the original URL and a comment "Based on this SO answer that does ...". However, someone else's code should make up only a small portion of any student's contribution.

## Extra Credit

Throughout the semester there may be opportunities to go above-and-beyond that will result in extra credit applied to the student's overall grade. These will be judged on a case-by-case basis, in a manner that is fair to the entire class. Examples of opportunities for extra credit include: exemplary effort in a particular iteration of the group project, presenting a research topic to the class, contributing to an in-class activity, or contributing to an open-source project.

## Grading

- 60% group project
  - 5% individual project setup
  - 15% group grades (e.g., runnable project and in-class activities)
  - 40% individual iteration contributions
- 40% quizzes
  - Can not make these up!

I will use the standard grading scale to convert to letter grades with plus and minus.

**The instructor reserves the right to revise, alter or amend this syllabus as necessary. Students will be notified regarding any changes.**