

CMPSC 481
Software Innovation 2
Spring 2020

If anything deserves a reward, it is social contribution. Creativity can be a social contribution, but only in so far as society is free to use the results.

—Richard Stallman, “The GNU Manifesto”

Course Information

Work 1:30 - 2:20 R Alden Hall 101

Instructor Douglas Luman
Email dluman@allegheny.edu
Office Alden Hall 105
Phone +1 814 332 2881

Student Hours

Monday 9:00 - 11:00; 3:30 - 5:00 (Doane Hall A104/106)
Wednesday 9:00 - 11:00; 3:30 - 5:00 (Doane Hall A104/106)
Thursday 9:00 - 12:00
Friday 9:00 - 11:00; 1:30 - 3:00

To schedule an appointment, visit <http://cs.allegheny.edu/sites/dluman>. Should none of the above times work for your schedule, let me know: we can work out a better time either via email or our course’s **Slack channel**.

Required Texts

This course requires a wide variety of reading; I will supply either links to or reproductions of any material assigned.

Canonical Course Description

A continuation of CMPSC 480, with a focus on the collaborative enhancement of innovative software. While partnering with members of the course and receiving mentorship from experts in the software community, students document, improve, and maintain a publicly available software tool. During a weekly practical session, students use state-of-the-art technology to complete and release a significant software product, describing it through oral presentations and written documents featured on a web site.

Prerequisite Knowledge	CMPSC 203 and CMPSC 480
Distribution Requirements	None

Another Course on “Software Innovation”?!

CMPSC 480 helped you learn skills centered on self-presentation. Though you will spend a fair bit of your career refining and growing these abilities, working in the disciplines directly and tangentially related to computer science often requires the real goods: projects.

While many developers credit involvement in open source projects as their true career launchpad, even those who successfully navigated the maze of “Pull Requests” and endless pages of release notes and discussion often admit that barriers to their first contribution were steep.

As a developer, you’ve undoubtedly used open source software. In fact, it’s guaranteed that software like your operating system and the Internet wouldn’t have been possible without it! This course, then, focuses on transforming you from a user to a contributor or, if you’re already part of a public project, growing your open source profile.

Learning objectives

By the end of this course, you will:

- Experience software development as a collaborative practice, cultivating collaboration skills
- Contributing to open source software projects, deepening your software development practice
- Connect with a large, vibrant open source community which often informally conducts mentorship and skills development
- Leverage these informal relationships to use your self-presentation skills and learn healthy development practices

But, let’s be fair: those are somewhat abstract goals. In the service of accomplishing these lofty aspirations, you will form teams and complete a variety of tasks that include (but are not limited to):

- Submit at least three (3) pull requests to open source software projects
- Rotate between various roles, giving them practice and experience developing, testing, and documenting one complete development cycle
- Present your experience working as a team in the open source ecosystem
- Practice GitHub Flow or other version control software (VCS) platform skills

Grading

Module 1	25% (250 pts.)
Module 2	25% (250 pts.)
Module 3	25% (250 pts.)
Final presentation	25% (250 pts.)
Total	100% (1000 pts.)

Class Participation: The major vehicle for class structure is your participation. This can mean attending class sessions, but also encompasses attending group meetings or contributing to group communication, digitally. You will receive four (4) participation scores through the semester: one

(1) for each module and one (1) for the final presentation. These scores will be the result of your and your group's assessment of your participation in the module and count toward individual module grades earned.

Modules: The semester is composed of three (3) modules, each running roughly one (1) month:

- Module 1: 1/16 - 2/13
- Module 2: 2/13 - 3/12
- Module 3: 3/12 - 4/16

“Modules” are defined periods of time during which students (organized into semester-long teams) contribute to a chosen open source project while fulfilling the obligations of one (1) of three (3) roles. Projects to which students will contribute may be team-selected and may persist for the entire semester (i.e. teams may contribute to the same project for all three modules).

Each student will occupy any of the following three roles *assuming a new one at the start of each module which they have not occupied in a previous module*:

- Developer
 - Students in the Developer role are responsible for implementing solutions to either bugs or feature requests documented by the team-chosen project's community. These students should implement and follow best practices for software engineering developed in CMPSC 203.

This role includes, but is not limited to:

- Developing code to solve an issue or implement a feature
- Conducting initial, shallow testing of developed code
- Providing adequate inline documentation per the standards of the community to which contribution is being made
- Collaborating with team members to implement and conduct thorough requirements-gathering

- Tester

- Students serving the Tester role develop and implement testing protocols for ensuring that developed code meets the needs of the original community request.

This role includes, but is not limited to:

- Guiding the requirements-gathering process
- Developing appropriate testing via best-suited methods (user stories, black box testing, unit testing, et al.)
- Conducting complete and thorough testing according to developed testing protocol
- Facilitating communication between the Writers and Developers

- Writer

- Writers manage external communication between the team and the chosen project. In addition, a student serving as a writer for a given module maintains the overall repository for the duration of a given module.

This role includes, but is not limited to:

- Corresponding with the chosen project’s community via the appropriate issue-tracking system
- Documenting overall progress for the team
- Compiling and submitting the module’s final technical report
- Creating documentation supporting the release of a “pull request” (this may mean README files or other community-standard documentation)

Each module will be concluded by submission of a Final Technical Report (FTR) which will require input from all team members *and* an outside mentor. The format for these reports will be given after students have chosen projects and begun work in the first module. It will be distributed through and should be turned in via GitHub.

Assignment Deadlines

Like post-academic deadlines, software developed for each module is due by the end date of the module. This means that the end dates listed above are *hard deadlines*.

However, you may finish a Pull Requests before that time, and choose to work on more. *As long as one (1) Pull Request of reasonable complexity is submitted during or at the end of a module, the guidelines of this course consider that module submitted.*

Note: The above does not contemplate whether the Pull Request is *accepted*. This is sometimes a “political football”, and (given that the contribution of the group is high-quality) the instructor cannot guarantee that the work will be accepted by the project.

Note on Attendance

All sessions will function as dedicated working time. Of course, this contemplates that team development processes don’t completely take place for only weekly, 50-minute intervals. Your instructor encourages you to meet outside of class.

Class attendance is mandatory. That written, you may encounter circumstances which require absence during a given class period. Generally, I require notification one (1) week in advance of an absence. Should you be absent due to an emergency, notify me as soon as is practical. Students who accumulate more than five (5) unexcused sessions will have their course grade reduced one (1) letter grade. Frequent or prolonged absences due to illness should be documented by the student’s doctor, the Health Center, the Dean of Students office, or the Student Disability Services.

Class Preparation

I expect students to arrive to class prepared. Here, “prepared” means being in communication and aligned with group development timelines and outstanding tasks. Students should be aware of and working toward common team goals during class periods. This can mean that teams may use class time for meetings or to do work.

Classroom Ethics

The discipline of computer science, like many others, encourages its members to act according to discipline-specific ethics. I encourage you to take time to review the Association for Computing Machinery (ACM) **Code of Ethics**. The Department of Computer Science maintains a brief poster version of this code on the first floor of Alden Hall.

The Allegheny College Statement of Community

Allegheny College also expects students and faculty to act according to its Statement of Community:

Allegheny students and employees are committed to creating an inclusive, respectful and safe residential learning community that will actively confront and challenge racism, sexism, heterosexism, religious bigotry, and other forms of harassment and discrimination. We encourage individual growth by promoting a free exchange of ideas in a setting that values diversity, trust and equality. So that the right of all to participate in a shared learning experience is upheld, Allegheny affirms its commitment to the principles of freedom of speech and inquiry, while at the same time fostering responsibility and accountability in the exercise of these freedoms. This statement does not replace existing personnel policies and codes of conduct.

Keep both of these standards in mind as you exercise your academic inquiry in this course. These serve as our fundamental “first principles” in pursuit of our shared academic goals.

Seeking Assistance

Assistance with course concepts

Students who struggle to understand knowledge and skills defined in this course are encouraged to seek assistance from the course instructor. To meet with me, consult my available office hours (above) and make an appointment.

tl;dr: historically, students who are successful in my courses visit and discuss questions and concerns about course concepts with the instructor early and often.

Assistance outside of the course

If you find yourself in difficult circumstances which affect your ability to participate in or complete course work, let me know immediately. Full stop.

Do not wait until the end of the semester. Part of my role as a course instructor is to make sure that students receive the assistance they need. Do not hesitate to let me know if there is any-

thing I can do with respect to your ability to handle your work.

In some situations, it may be helpful to consider the following resources:

The Maytum Learning Commons +1 814-332-2898
<http://sites.allegheny.edu/learningcommons/tutoring/>

Allegheny College Counseling Center	Schultz Hall	+1 814-332-4368
The Winslow Health Center	Schultz Hall	+1 814-332-4355
Student Life	Campus Center	+1 814-332-2800

Special Needs and Disability

Students with disabilities who need accommodations in this course are encouraged to contact Disability Services at +1 814-334-2898. Disability Services is part of the Learning Commons, located in Pelletier Library. Should you need accommodations, contact this office as soon as possible to ensure that approved accommodations are communicated and implemented as quickly as possible. This serves both you and I in providing the best environment for learning and support.

Honor Code

All students and faculty at Allegheny College are bound by the Honor Code. Everyone expects that your behavior reflects this commitment. Given the eminently shareable and reproduceable nature of code, the Department of Computer Science adds the following statement to the general college policy:

Department of Computer Science Honor Code Policy

It is recognized that an important part of the learning process in any course, and particularly in computer science, derives from thoughtful discussions with teachers, student assistants, and fellow students. Such dialogue is encouraged. However, it is necessary to distinguish carefully between the student who discusses the principles underlying a problem with others, and the student who produces assignments that are identical to, or merely variations on, someone else's work. It will therefore be understood that all assignments submitted to faculty of the Department of Computer Science are to be the original work of the student submitting the assignment, and should be signed in accordance with the provisions of the Honor Code. Appropriate action will be taken when assignments give evidence that they were derived from the work of others.

As the nature of "plagiarism" and constituents of "fair use" change often, the department encourages you to periodically review the specific tenets of the general college Honor Code provided in the latest course catalog and in the *Compass*.

The above statement, of course, also applies to online forums such as Stack Overflow, et al.