

INST 447
Data Sources and Manipulation
Spring 2018
Syllabus

Course Instructor

Mr. Donal Heidenblad
Office Location: South Hornbake 4121F
Email: dheidenb@umd.edu
Twitter: @donal_h
Web Site: <http://www.donal.us/>

Instructor's Office Hours

- Monday: 2:30 pm–3:30 pm (15 minute time slots)
- Thursday: 11:00 pm–12:00 noon (15 minute time slots)

To schedule a meeting with me during my office hours, please visit my web site and click the “Schedule” link in the top right-hand corner. Now, you can browse my office hours or schedule an appointment by clicking the correct link and then reserving an open time slot. Students are also encouraged to post appropriate questions to a relevant channel in our Slack team, which is available at <https://CMPSC280Fall12017.slack.com/>, and monitored by the course instructor.

Course Meeting Schedule

Lecture, Discussion, and Group Work: Monday, Wednesday, Friday, 1:30 pm–2:20 pm
Laboratory Session: Tuesday, 2:30 pm–4:20 pm
Final Examination: Friday, December 15, 2017 at 7:00 pm

Course Description

A human-centric study of the principles used during the engineering of high-quality software systems. In addition to examining the human behaviors and social processes undergirding software development methodologies, students participate in teams tasked with designing, developing, and delivering a significant software application for a customer. During a weekly laboratory session, students use state-of-the-art software engineering, management, and communication tools to complete projects, reporting on their results through both written reports and oral presentations. Prerequisite: CMPSC 112. Distribution Requirements: SB, SP.

Course Objectives

The process of developing software involves the application of a number of interesting theories, tools, techniques, and methodologies. In this class we will explore the phases of the software engineering life cycle and examine the tools, concepts, challenges, and open questions associated with each phase. Throughout the semester, we will investigate the interplay between the theory and practice of software engineering. Specifically, we will delve into the details of software specification, design, implementation, testing, and maintenance through a discussion of book chapters and articles from

the software engineering and software testing literature. Along with learning more about how to effectively work in a team of diverse software developers, students will enhance their ability to write and present ideas about software in a clear, concise, and compelling fashion. Students will develop an understanding of the fascinating connections between computer science and software engineering and other disciplines in the social and natural sciences and the humanities. Students also will gain software engineering experience when completing laboratory assignments and a final project.

Performance Objectives

At the completion of this class, a student should be aware of the fundamental challenges associated with software engineering. Furthermore, students should be comfortable with a wide array of concepts, methodologies, techniques, and tools that they apply to the problem of developing large software systems. However, a successful student will emerge with more than an understanding of the tools (e.g., text editors, compilers, debuggers, integrated development environments, and version control systems) that a software engineer uses. A student also should have a fundamental understanding of the software engineering life cycle and the activities that take place in each of its phases. Finally, a student should have a basic understanding of some of the current research and the open questions in the field of software engineering. After completing this class, a student should be equipped for further graduate study in the fields of computer science and software engineering. The student should also be able to participate in real-world software development projects by adeptly using cutting-edge software tools and working with a team of diverse developers.

Required Textbooks

Software Engineering: Theory and Practice. Shari Lawrence Pfleeger and Joanne M. Atlee. Fourth Edition, ISBN-10: 0136061699, ISBN-13: 978-0136061694, 792 pages, 2010.

(References to the textbook are abbreviated as “SETP” in the syllabus and on the web site).

The Mythical Man Month. Frederick P. Brooks, Jr. Second Edition, ISBN-10: 0201835959, ISBN-13: 978-0201835953, 336 pages, 1995.

(References to the textbook are abbreviated as “MMM” in the syllabus and on the web site).

Students who want to improve their technical writing skills may consult the following books.

BUGS in Writing: A Guide to Debugging Your Prose. Lyn Dupré. Second Edition, ISBN-10: 020137921X, ISBN-13: 978-0201379211, 704 pages, 1998.

Writing for Computer Science. Justin Zobel. Second Edition, ISBN-10: 1852338024, ISBN-13: 978-1852338022, 270 pages, 2004.

Along with reading the required books, you may be asked to study many additional articles from a wide variety of conference proceedings, scientific journals, and the popular press.

Course Policies

Grading

The grade that a student receives in this class will be based on the following categories. All percentages are approximate and, if the need to do so presents itself, it is possible for the course instructor to change the assigned percentages during the academic semester.

Class Participation	5%
SEED Project Participation	5%
First Examination	15%
Second Examination	15%
Final Examination	20%
Laboratory Assignments	25%
Final Project	15%

These grading categories have the following definitions:

- *Class Participation*: All students are required to actively participate during all of the course sessions. Your participation will take forms such as answering questions about the reading assignments, asking constructive questions of group members, giving presentations, and leading a discussion. You also must regularly participate in the discussions in the course's Slack team. A student may request feedback on and will receive a final grade for this category.
- *SEED Project Participation*: As part of an ongoing project that publishes educational discussions with software engineers, every week I will publish, on my blog available at <http://www.cs.allegheeny.edu/sites/gkapfham/blog/>, an interview with a software engineer in industry. At the start of the laboratory session on Tuesday of each week, every student will read the blog post and tweet a response to that week's interview. Along with linking to the web site of the blog post and using the hashtag #SEED, your tweet should share what you learned from the interviewee. Interested students should also connect with the interviewee through LinkedIn.
- *First and Second Examinations*: The first and second interim examinations will cover all of the material in their associated module(s), as outlined on a review sheet. While the second examination is not cumulative, it will assume that a student has a basic understanding of the material that was the focus of the first examination. The date for the first and second examinations will be announced at least one week in advance of the scheduled date. Unless prior arrangements are made with the course instructor, all students will be expected to take these examinations on the scheduled date and complete the tests in the stated period of time.
- *Final Examination*: The final examination is a three-hour cumulative test. By enrolling in this course, students agree that, unless there are extenuating circumstances, they will take the final examination at the date and time stated on the first page of the syllabus.
- *Laboratory Assignments*: These assignments invite students to explore the concepts, tools, and techniques that are associated with different phases of the software engineering life cycle. All of the laboratory assignments require the use of the provided tools to design, implement, test, maintain, and document programming system products that solve important problems. To ensure that students are ready to develop software in both other classes at Allegheny College and after graduation, the instructor will assign individuals to teams for each of the laboratory assignments. Unless specified otherwise, each laboratory assignment will be due at the beginning of the next laboratory session. Many of the course's laboratory assignments will expect students to give both an oral presentation and an interactive demonstration of the software that they correctly specified, designed, implemented, tested, and documented. When teamwork is required, the instructor will often assign individuals to their teams.
- *Final Project*: This project will present you with the description of a problem and ask you to implement, document, and release a full-featured solution. The final project in this class will require you to apply all of the knowledge and skills that you have acquired during the semester to solve a problem and, whenever possible, make your solution publicly available as a free and open-source tool. The project will invite you to draw upon both your communication and

problem solving skills and your knowledge of programming languages and software engineering tools. The final project will be completed in groups assigned by the course instructor.

Assignment Submission and Evaluation

All assignments will have a stated due date. Electronic versions of the laboratory and final project assignments must be submitted to a student's GitHub repository; students will review how to use version control with GitHub during the first laboratory session. No credit will be awarded for any course work that is not submitted to your GitHub repository with the required name. Unless specified otherwise, all of assignments must be turned in at the beginning of the session that is one week after the day the assignment was released. If you do not make special arrangements with the course instructor, no work for an assignment will be accepted after the published deadline.

Using a report that the instructor shares with you through the commit log in GitHub, you will privately receive a grade for and feedback on each assignment. Your grade will be a function of whether or not you completed correct work and submitted it by the deadline. Other factors (e.g., the quality of your source code and technical writing) will also influence your assignment grade. For every assignment that requires team work, the instructor will consider, when assigning grades, the individual contributions of each team member and the team's overall ability to work together.

Course Attendance

It is mandatory for all students to attend the class and laboratory sessions and all group project meetings. If, due to extenuating circumstances, you will not be able to attend one of these events, then, whenever possible, please see the course instructor at least one week in advance to describe your situation. Students who miss more than five unexcused classes, laboratory sessions, or group project meetings will have their final grade in the course reduced by one letter grade. Students who miss more than ten of the aforementioned events will automatically fail the course.

Use of Laboratory Facilities

Throughout the semester, we will investigate many different tools that developers use during the phases of the software engineering life cycle. Students must leverage these tools to complete all laboratory assignments and the final project while using the department's laboratory facilities.

Class Preparation

In order to minimize confusion and maximize learning, students must invest time to prepare for class discussions and lectures. During the class periods, the course instructor will often pose demanding questions that could require group discussion, the creation of a program or test suite, a vote on a thought-provoking issue, or a group presentation. Only students who have prepared for class by reading the assigned material and reviewing the current assignments will be able to effectively participate in these discussions. More importantly, only prepared students will be able to acquire the knowledge and skills that are needed to be successful in both this course and the field of software engineering. In order to help students remain organized and effectively prepare for classes, the instructor will maintain a class schedule with reading assignments and presentation slides.

Seeking Assistance

Students who are struggling to understand the knowledge and skills developed in a class or laboratory session are encouraged to seek assistance from the course instructor. Throughout the semester,

students should, within the bounds of the Honor Code, ask and answer questions on the Slack team for this course; please request assistance from the instructor first through Slack before sending an email. Students who need the course instructor's assistance must schedule a meeting through his web site and come to the meeting with all of the details needed to discuss their question.

Using Email

Although we will primarily use Slack for class communication, the instructor will sometimes use email to send announcements about important matters such as changes in the schedule. It is your responsibility to check your email at least once a day and to ensure that you can reliably send and receive emails. This policy is based on the statement about email use that appears in *The Compass*, the College's student handbook; please see the instructor if you do not have this handbook.

Disability Services

The Americans with Disabilities Act (ADA) is a federal anti-discrimination statute that provides comprehensive civil rights protection for persons with disabilities. Among other things, this legislation requires all students with disabilities be guaranteed a learning environment that provides for reasonable accommodation of their disabilities. Students with disabilities who believe they may need accommodations in this class are encouraged to contact Disability Services at 332-2898. Disability Services is part of the Learning Commons and is located in Pelletier Library. Please do this as soon as possible to ensure that approved accommodations are implemented in a timely fashion.

Honor Code

The Academic Honor Program that governs the entire academic program at Allegheny College is described in the Allegheny Academic Bulletin. The Honor Program applies to all work that is submitted for academic credit or to meet non-credit requirements for graduation at Allegheny College. This includes all work assigned for this class (e.g., examinations, laboratory assignments, and the final project). All students who have enrolled in the College will work under the Honor Program. Each student who has matriculated at the College has acknowledged the following pledge:

I hereby recognize and pledge to fulfill my responsibilities, as defined in the Honor Code, and to maintain the integrity of both myself and the College community as a whole.

It is understood that an important part of the learning process in any course, and particularly one in computer science, derives from thoughtful discussions with teachers and fellow students. While it is acceptable for students in this class to discuss their work with their classmates, deliverables that are nearly identical to the work of others will be taken as evidence of violating the Honor Code.

Welcome to a Software Engineering Adventure

In reference to software, Frederick Brooks, Jr. wrote in Chapter One of MMM, "The magic of myth and legend has come true in our time." Software is a pervasive aspect of our society that changes how we think and act. High quality software also has the potential to positively influence the lives of people. Moreover, the specification, design, implementation, testing, maintenance, and documentation of software are exciting and rewarding activities! At the start of this class, I invite you to pursue, with great enthusiasm and vigor, this adventure in software engineering.