

CS 362 – Software Engineering II Syllabus

Course Description

In this course we will be discussing Software Testing. We will cover what software testing is and where it fits into the software design process. We will also be exploring many different approaches to testing and for which applications each is best suited. This is not a traditional programming class. You will be expected to fill any gaps in your programming knowledge to accomplish the assigned tasks.

- Prerequisites: CS 261
- Credits: 4
- Terms Offered: Every Term

Instructor

Eric Ianni

Email: iannie@oregonstate.edu

Communication Policy

- **Always use your OSU email to contact us. The Canvas mailbox doesn't work very well.**
- **When you send us an email, you must include the tag "[CS 362]" in your email subject.**

Please use the email addresses above to contact the instructor and TAs. You should expect a response to emails within 48 hours. Emails sent over the weekend sometimes take longer to respond to.

Post all course-related questions on the Piazza board so the whole class may benefit from our conversation. Please sign up yourself on Piazza. You can use the course Slack channel to ask questions as well.

For grading questions and regrading request, please contact the TA who graded your assignment by posting privately on Piazza to the "instructors" and tagging the TA in the post. Don't post re-grading request on Slack. You should expect your grade to be posted after one week of the due date. If you submit the assignment late, it may take longer for your grade to be released.

The instructional team will be using the class mailing list extensively to communicate with you. We will also frequently post information on Piazza. It is your responsibility to keep up-to-date with these communiques and they are considered part of the required learning material.

Course Topics

- Version Control Systems such as Git
- Software Verification

- Unit Testing
- Black Box Testing
- White Box Testing
- Random Testing
- Test Driven Development
- Continuous Integration
- Code Review

Measurable Student Learning Outcomes:

At the completion of the course, students will be able to:

1. **Apply** automated tools such as `_make_` and CVS in a realistic setting
2. **Describe** the cost-benefit trade-offs inherent in the use of automated tools for building software and configuration management
3. **Describe** several techniques for validating and measuring the quality of software
4. **Apply** testing techniques for validating and measuring the quality of software
5. **Use** appropriate techniques and tools, including a debugger, to locate program faults
6. **Describe** several types of maintenance processes associated with correcting and enhancing software systems
7. **Participate** effectively in a software inspection/code review
8. **Participate** effectively in a team environment

Course Schedule

Topics by Weeks

WEEK	TOPICS	DUE
1	Tool Setup What is Testing	Quiz: Course Policy A2: Git Quiz: Testing
2	Black Box Testing	HW1: Writing Black Box Tests
3	White Box Testing	HW2: Improving Coverage
4	Random Testing	HW3: Random Testing Hands On
5	Test Driven Development	A2: TDD Hands On
6	Continuous Integration Code Review	Group Project: Part 1 A3: Code Review Hands On
7	Stubs, Fakes, and Mocks	Group Project: Part 2 Group Project: Part 3 Quiz: Testing Environments and Mocking
8	Advanced Topics	Practice Final Exam Final Exam

Textbooks

There are no required textbooks for this course

Assignments

This course has 5 types of assignments:

- Activity: These are worth the least amount of points and generally are looking for general topic understanding.
- Homework: These are graded for correctness and require more work than an activity. They are also worth more points.
- Quiz: These are straightforward multiple-choice assessments. You have two attempts for each with the highest score kept.
- Project: This is a group effort and will span multiple weeks. It is the assignment worth the most points excluding the final.
- Final Exam: This is an all-encompassing assessment. Any topic discussed in the course materials is eligible. It will consist of multiple-choice, short answer, and simple coding questions.

Exam

This course has one proctored exam -- the final exam. You can find out more about proctoring at the central Ecampus page on [tests and proctoring](#).

The final exam window will run from the **Sunday** at the start of finals week through **Thursday** of finals week. If you are unable to take the exam in that window, you must make arrangements **prior to the end of the 2nd week** of classes. Beyond this deadline, only emergency situations will be considered for alternate testing times.

Where possible, I suggest using an in-person proctor. Should an issue arise, it is historically a lot easier to get it resolved at a testing center than with other online proctoring service. If you do use ProctorU and an issue does arise, please document the situation as thoroughly as possible and forward that to the instructor as soon as possible.

Grading Policy

Grade letter	Percentage floor
A	93
A-	90
B+	87
B	83
B-	80
C+	77
C	73
C-	70
D+	67
D	63
D-	60
F	0

Grade Weighting

There are no weighted categories for assignments. Each assignment is assigned a given number of points. The final grade percentage will be determined using $\frac{\text{Total points received}}{\text{Total points possible}}$

Accommodations

"Accommodations are collaborative efforts between students, faculty and Disability Access Services (DAS). Students with accommodations approved by DAS are responsible for contacting the faculty member in charge of the course prior to or during the first week of the term to discuss accommodations. Students who believe they are eligible for accommodations but who have not yet obtained approval through DAS should contact DAS immediately at 541-737-4098."

Students with documented disabilities who may need accommodations, who have any emergency medical information the instructor should be aware of, or who need special arrangements in the event of an evacuation, should make an appointment with the instructor **as early as possible**, and no later than the first week of the term. Class materials will be made available in an accessible format upon request.

If you have a really tough situation that might affect your progress a lot (illness, job duties, family emergency...), you should contact the instructor immediately. **Don't wait until the due date or even past the due date to explain your personal situations and ask for extensions.** If you are not sure whether to ask for it, better do it.

Late Policy

Requests for extensions are considered on a case by case basis. Non-emergency requests must be submitted **via email at least 72 hours before the due time**. (Not having enough time to get the assignment done does not, by itself constitute an emergency, sorry!). **If you don't know if you will need an extension but might, you should ask for one.**

Time elapsed past due date	Penalty Applied (if no extension is granted)
T < 24 hours	-10%
T < 48 hours	-20%
T < 72 hours	-30%

Extra Credit

There is extra credit on some of the assignments. Try to get them when possible. At the end of the term, I will give extra credit (0.5 to 2 maximum points) to those who are active on Piazza answering questions and sharing notes with others.

Code Sharing

You will not get in trouble for sharing code with your classmates in order to solve problems. The communication guide actually mandates that you share **portions of your code** if you want to ask a good question. If you are worried that you are posting too much code, mark it private and ask the instructor to review it. Note that this is a more permissive policy than the [standard policy for the program](#).

You will get a great deal of trouble if you copy code without citing it. See the policy on [plagiarism](#). Code from lectures is not your own, code from StackOverflow is not your own, code from the Mozilla documentation is not your own. If it is not your code, you must cite it. If you cite it, you must provide documentation in very great detail of what it is doing so that I know you understand the code you are using.

Academic Misconduct

The Code of Student Conduct prohibits Academic Misconduct and defines it as:

Any action that misrepresents a student or group's work, knowledge, or achievement, provides a potential or actual inequitable advantage, or compromises the integrity of the educational process.

To support understanding of what can be included in this definition, the Code further classifies and describes examples of Academic Misconduct, including cheating, plagiarism, assisting and others. See the [Code of Student Conduct](#) for details.

You are expected to do your own work and demonstrate academic integrity in every aspect of this course. Familiarize yourself with the standards set forth in the OSU Code of Student Conduct Section 4.2. You must only access sources and resources authorized by the instructor. You may not show your work to any other current or future students without the instructor's authorization. Violations of these expectations or the Code of Student Conduct will be reported to the Office of Student Conduct and Community Standards. If there is any question about whether an act constitutes academic misconduct, it is your responsibility to seek clarification and approval from the instructor prior to acting.

Permission to use your code/submissions

By taking this course you acknowledge that your work may be shared with your peers for educational purposes. These purposes include, but are not limited to, peer review and code samples.