

* **Subject & Course: CS362 – Software Engineering II**

* **Credits: 4**

* **Professor's name: Alex Groce**

* **Instructor's email and/or phone: agroce@gmail.com**

♦ <http://www.cs.cmu.edu/~agroce> has my most maintained web presence, but OSU has profiles, etc. My office hours are Tue/Thur 1:00-2:00 and by appointment

♦ **Teaching Assistant names and contact info:**

Xin Liu (liux4@oregonstate.edu)

Sahar Alizadeh (alizades@oregonstate.edu)

♦ **Teaching Assistant Office Hours:**

Xin Liu: Tuesday and Friday, 4:00-5:00pm

♦ **Syllabus effective for term(s): Spring 2016**

* **OSU catalog course description, including pre-requisites/co-requisites:**

Introduction to the "back end" of the software engineering lifecycle implementation; verification and validation; debugging; maintenance. PREREQS: CS 261 and Experience with object-oriented programming and data structures (e.g. CS 161, CS 162, CS 261). CS 361 is recommended but not required.

* **Course content: [Concise outline of topics and activities; course timeline by the week of the term, omitting specific dates.]**

♦ **Content** — This course will be "delivered" mostly via a github repository. You will need a google account! The project name is cs362sp16core for the materials. You can check it out to get all class material.

Course Outline:

0. Overview (Civil Engineering and Software Engineering)
1. Thinking about Testing
2. Maintenance and Source Control
3. Builds & Static Analysis
4. Introduction to Software Testing
5. Kinds of Testing (Manual vs. Automated; Scripted vs. Exploratory; Unit Testing; Integration Testing; System Testing; Regression Testing; Black Box vs. White Box)
6. How Tested Is It? Coverage Metrics I
7. Coverage Metrics II
8. Lessons Learned in Software Testing: The Testing Role

9. Lessons Learned in Software Testing: Thinking Like a Tester
10. Lessons Learned in Software Testing: Testing Techniques
11. How to Write a Simple Random Tester
12. Random Testing: Not Just For Toys
13. Lessons Learned in Software Testing: Reporting Bugs and Working Well With Others
14. Lessons Learned in Software Testing: Planning and Strategy
15. Introduction to Debugging
16. Quick Intro to Debuggers
17. Causality and Localization I
18. Causality and Localization II
19. Agans' Rules for Debugging
20. Software Inspections & Course Summary

*** Measurable student learning outcomes:** [What will students learn in your course and how you will verify this learning. See [Student Learning Outcomes](#) for a definition and instructions.]

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

- Apply automated tools such as make and CVS in a realistic setting
- Describe the cost-benefit trade-offs inherent in the use of automated tools for building software and configuration management
- Describe several techniques for validating and measuring the quality of software
- Apply testing techniques, including black-box and white-box techniques, automatic testing activities, and regression testing
- Use appropriate techniques and tools, including a debugger, to locate program faults
- Describe several types of maintenance processes associated with correcting and enhancing software systems
- Participate effectively in a software inspection
- Participate effectively in a team environment

*** Learning resources:** [Textbooks, lab kits, streaming media, course packets, etc.]

*The following are instructive books, but are **NOT REQUIRED**. There is no required text.*

Resources

- *Lessons Learned in Software Testing*, by Cem Kaner, James Bach, and Bret Pettichord;
- *Debugging* by David J. Agans

*** Evaluation of student performance:** [How will the learning outcomes will be measured (exams, projects, discussions, etc)? What is your grading scale?]

Scores for quizzes, assignments, and exams will be posted on Blackboard as they are graded.

Assignments - 40%

- There are 4 total assignments to be completed over the course of this class.
- Assignments include a mixture of written documents and code submissions.
- Assignments are all submitted via checkin to your github repository for the class!

- Advice for github: remember, **add**, **commit**, **push**
E.g., if you write mycrazydominioncode.c and .h you need to:
> **git add mycrazydominioncode.[ch]**
> **git commit -m "Added my crazy dominion code!"**
> **git push**
- If you have a problem with an assignment grade, you must contact the teaching assistant, who graded your assignment, through EMAIL within ONE WEEK of receiving your grade.

Exams - 20% (20% each exam)

- There is 1 exam for this course.
- Each exam is given after completing 20 class lectures.
- These tests are designed to take 60-90 minutes each.
- These exams are open note, open internet essay exams.
- Rule: the “midterm” will be given at least one week after I give notice, and after the 20th mini-lecture above has been completed.

Final Project - 40%

- There is a final project designed to check for a cumulative understanding, which includes some of the work for assignments.
- For this project, you will work alone, but your work will be focused on your classmates’ code.
- Like assignments, all submission is via github.
- In the past, students find the final project a considerable step up in difficulty over the other assignments, so get started early.

Grading Scale

Grade	Average
A	93 or greater
A-	90 - 92
B+	87 - 89
B	83 - 86
B-	80 - 82
C+	77 - 79
C	73 - 76*
C-	70 - 72
D+	67 - 69
D	63 - 66
D-	60 - 62
F	less than 60

* REMINDER: A passing grade for core classes in CS is a C or above. A C-, 72 or below, is not a passing grade for CS majors.

Academic Dishonesty

I encourage students to work together and learn from one another on labs and assignments. However, I do expect you to turn in your OWN work. Working with someone does not include copying someone else's work and changing a small amount of that work, such as variable names, comments, spacing, etc. During group assignments you and your partners may turn in one assignment per group with everyone's name attached. Working together is discouraged on exams, quizzes, and the final. At NO point should you copy work from the internet, and if you do copy material from an external resource, then you need to cite the resource and author(s). Cheating and plagiarism are not taken lightly! You will receive a zero on your first abuse of these rules. In the case of shared work, the student sharing the work and the student copying the work will both receive zeros. On the second abuse, your name(s) will be given to the EECS department, where they will handle the details. Please read the [department](#), [college](#), and [university](#) dishonesty policies.

Students with Disabilities

"Accommodations are collaborative efforts between students, faculty and Disability Access Services (DAS). Students with accommodations approved through 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 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.

Expectations for Student Conduct:

Student conduct is governed by the university's policies, as explained in the [Office of Student Conduct: information and regulations](#).

Additionally, programming assignments in this course are considered Take Home Programming Tests. You must do your own work, entirely.

- You **MAY** discuss the meaning of assignments, general approaches, and strategies with other students in the course.
- You **MAY** show your code to the TAs or instructor for feedback and help.
- You **MAY** use the Internet to research how to solve a problem.
- You **MUST** include a citation in the form of a comment in your source code to indicate the source of any help you received (except the TAs).

- You **MUST ALSO** include a citation if you collaborated with any other student in any way (both the giver and receiver).
- You **MAY NOT** show your assignment code to another student in the course for any reason.
- You **MAY NOT** use or copy code from any other source, including the Internet.
- You **MUST** write your own code for your assignments.

We use plagiarism-detection software check your code against the code from other students. It is quite sophisticated and can easily see through variable name changes and formatting differences.

If you are found in violation of any of the above policies, whether you are the giver or receiver of help, you will receive a zero on the assignment or fail the course (Instructor's discretion). The academic dishonesty charge will be documented and sent to your school's dean and the Office of Student Conduct. The first offense results in a warning; the second offense results in an academic dishonesty charge on your transcript, a disciplinary hearing, and possible expulsion.

♦ Student Assistance:

♦ **Technical Assistance** — If you experience computer difficulties, need help downloading a browser or plug-in, assistance logging into the course, or if you experience any errors or problems while in your online course, contact the OSU Help Desk for assistance. You can call (541) 737-3474, email osuhelpdesk@oregonstate.edu or visit the [OSU Computer Helpdesk](#) online.

♣ Tutoring —

Effective fall term 2009 we went to a new Online Tutoring Service - [NetTutor](#) to meet the needs of Ecampus students.

NetTutor is a leading provider of online tutoring and learner support services fully staffed by experienced, trained and monitored tutors. Students connect to live tutors from any computer that has Internet access. NetTutor provides a virtual whiteboard that allows tutors and students to work on problems in a real time environment. They also have an online writing lab where tutors critique and return essays within 24 to 48 hours.

♣ **Course Evaluation:** [Some instructors provide a discussion board for ongoing student comments and suggestions during the term. Often they choose to allow anonymous posts. Soliciting input from students about the course during the term is valued by students and gives instructors the opportunity to deal with problems while the current batch of students can still benefit.]

♣ **OSU Student Evaluation of Teaching** — Course evaluation results are extremely important and are used to help me improve this course and the learning experience of future students. Results from the 19 multiple choice questions are tabulated anonymously and go directly to instructors and department heads. Student comments on the open-ended questions are compiled and confidentially forwarded to each instructor, per OSU procedures. The online Student Evaluation of Teaching form will be available toward the end of each term, and you will be sent instructions by Ecampus. You will login to "Student Online Services" to respond to the online questionnaire. The results on the form are anonymous and are not tabulated until after grades are posted.