

SOFTWARE ENGINEERING CONCEPTS

ECE473-573

Jonathan Sprinkle

Day/Time: MW 4:00-5:15 pm
Location: [ECE 107](#)

<http://www.ece.arizona.edu/~ece473/>

Instructor: Jonathan Sprinkle
ECE 456N
Email: sprinkle@ECE.Arizona.Edu

Office Mon 10:30-11:30, Wed 1:30-2:30, and
Hours: by appointment or email.

Reference(s): *Software Engineering*. I. Sommerville. 9th Ed. Addison-Wesley. 2010. ISBN-13: 978-0137035151
The SVN Book, <http://svnbook.red-bean.com/>
Additional papers and references, as assigned.

Prerequisites: ECE 373 or equivalent, or consent of instructor prior to early withdrawal date. Graduate enrollment requires graduate standing, or written consent of the instructor. Auditing requires written consent of instructor, prior to audit change date. You cannot sit in on this class without registering as an auditor.

Course Description:

From the course catalog: In-depth consideration of each of the phases of the software project life code [sic]. Object-oriented design and programming. Includes a large-scale software development project involving groups of students. Graduate-level requirements include additional homework and a term project.

This course will teach you the relevant phases of medium- and large-scale software design. As such, it is a departure from standard software courses taught in a university setting, since a medium or large software project that is self-contained and fully original will be developed by successful students.

The course will provide a relevant experience (similar to what would be experienced in industry) in medium- and large-scale software development and maintenance, through several key aspects:

- students will learn, on their own, how to use an existing software framework, as decided by the instructor;
- student projects will be integrated with existing open-source software;
- projects and assignments will utilize engineering concepts, and may interface with hardware;
- requirements will be subject to modification by the instructor during the course of the term;
- documentation and usage will constitute a major portion of the term grade.

In course homework, as well as in your project, you will use programming exercises as proofs of concept. This is *not* primarily a programming course: this does not mean you will not program, but it *does* mean that you are expected to already know *or be willing to learn* Java, C++, Objective-C, C#, or some other object-oriented language. If you do not know how to program in any *object-oriented* language, see the instructor *immediately* after class to discuss any options. Students who desire to have programming exercises to review may see the website.

Academic Topics

The set of topics and areas covered by this course, and upon which you may be tested, include:

- code lifecycle, including conceptualization, analysis, design, implementation, and maintenance;
- software development processes, including waterfall and iterative processes, the UML methodologies, and CMMI;
- formal methods, including model-driven designs, requirements specifications, automata theory;
- software metrics, including code coverage;
- code review processes; and
- certification of software.

Technical Topics

Technical processes and skills of which you will gain knowledge of in this course include:

- code releases;
- revision control systems;
- project configuration and management;
- testing, including regression testing; and
- code licensing options.

In addition to these topics, students who successfully complete the course will engage in a project which is approved by the instructor, and which is performed by a team (not an individual) for undergraduate students, and as an individual, for graduate students.

Important Dates:

R 1/23	Last day to add classes for credit from zero units
T 2/11	Last day to drop courses resulting in deletion of course enrollment from record
T 3/11	Last day to drop a class with a grade of “W” (if passing) or to change to or from audit grading; the instructor’s signature on a Change of Schedule form is required
W 5/7	Last day of classes and laboratory sessions
W 5/14	3:30-5:30 pm, Final Examination

Course Outline:

The listing of weekly course lecture topics may be found on the webpage, and is subject to change without notice due to class progress. In the event of class cancellation, advance notice *via email* will be given, but any homeworks due that day will still be due unless otherwise notified via email.

Grade Policy:

Homework:	15%	[90 – 100]	A
Examinations:	20%	[80 – 90)	B
Project:	50%	[70 – 80)	C
Attendance/Participation/Quizzes:	15%	[60 – 70)	D
		[0 – 60)	E

Assignment of grades is done according to a “modified-contract” method. The above scale represents a minimum guarantee. However, the instructor reserves the right to “upward curve” the final grade of the entire class, or of one or more individuals whose objective performance improves as the term progresses.

Students with Disabilities:

If you anticipate issues related to the format or requirements of this course, please meet with the instructor to discuss ways to ensure your full participation in the course. If you determine that formal, disability-related accommodations are necessary, it is very important that you be registered with the Disability Resource Center (621-3268; <http://drc.arizona.edu>) and afterward notify the instructor of your eligibility for reasonable accommodations. Only after that point can we plan how to best coordinate any accommodations.

Homework Companion

The homework must be submitted according to the guidelines set forth by the Homework Companion (available from the course webpage). Failure to abide by the Homework Companion may result in a failing grade, regardless of the correctness of the work.

Project Companion:

Students in the course will perform work on a course project which will be of significant value as deemed appropriate by the instructor. The course project will be submitted via the terms of the Project Companion, which will be distributed as described in the Schedule of Weeks.

Attendance, Participation, and Quizzes:

Attendance is mandatory. Although the class roll may not be taken every day, pop quizzes *may* be given without notice. Pop quizzes may not be made up, though a certain number may be dropped. In-class exercises will serve as objective measures of your participation and attendance, and may not be made up. Expected absences for valid reasons (e.g., travel to a conference) *must be cleared in advance* to avoid penalty for missing participation.

Class Disruptions:

Please silence your cell phone, and do not use it during the class. The use of a phone in class will adversely affect your attendance grade*.

*Uh, um, unless you are programming it as part of an in-class exercise. But please no talking or texting. Unless that is part of the class exercise too.

Academic Integrity:

Students are expected to do all work by themselves, except when specified by the instructor in writing. All exceptions will be plainly marked in the requirements for that exercise or project. Any violations of this policy will be dealt with to the full extent permitted by the University of Arizona, and *may result in suspension or expulsion from the university, in addition to a failing grade*. Please familiarize yourself with the Code of Academic Integrity if you have any questions (see <http://deanofstudents.arizona.edu/codeofacademicintegrity>).

Safety Instructions:

The frequent operation of a computer, such as will be required in this course, may have long-term disabling effects if you do not appropriately consider your ergonomic interaction with the computer, desk, chair, and light sources. Poorly designed work stations/practices can lead to musculoskeletal disorders, and may result in chronic pain, inability to sleep, or expensive surgery decades from today. The habits you form in your university years may well impact your future performance, and it is highly recommended that you consult the free, online, ergonomics information from the Office of Risk Management, available at <http://risk.arizona.edu/healthandsafety/ergonomics.shtml>

ABET Classifications

Learning Outcomes

Individual Outcomes

By the end of this course, the student will be able to:

1. Discuss the phases of the software engineering process;
2. Identify requirements as functional, or nonfunctional;
3. Identify software analysis techniques as static, or dynamic;
4. Perform a design criticism, including design tradeoffs;
5. Develop engineering software for mechanical or electrical system simulation through object-oriented methods;
6. Understand technical issues regarding software development processes, including tradeoffs for different applications and the most appropriate SDP for an application;
7. Use version control software, and describe the fundamental differences between Lock-Modify-Unlock and Copy-Modify-Merge revision control systems;
8. Describe the fundamental types of testing available in the software design process;
9. Perform rudimentary software analysis calculations, including cyclomatic complexity, Halstead complexity, etc.;

Project Outcomes

In addition to the previous learning outcomes, the following project outcomes are anticipated:

10. use an existing open-source or licensed software package, as decided by the instructor;
11. integrate software with an existing open-source software package;
12. learn on her own the SDK and best practices for an open-source software package;
13. write software that utilizes electrical and mechanical engineering concepts, including an interface with hardware; (in this course, a mobile device)
14. specify requirements for a software design, in a design document;
15. modify an existing software design when requirements are changed by the customer (instructor); and
16. enforce consistent documentation and style through code reviews.

Program Outcomes

The following program outcomes are satisfied by this course:

- (a) an ability to apply knowledge of mathematics, science, and engineering (HIGH)
- (c) an ability to design a system, component, or process to meet desired needs (HIGH)
- (d) an ability to function on multi-disciplinary teams (MEDIUM)
- (e) an ability to identify, formulate, and solve engineering problems (HIGH)
- (g) an ability to communicate effectively (HIGH)
- (i) a recognition of the need for, and an ability to engage in life-long learning (MEDIUM)
- (j) a knowledge of contemporary issues (HIGH)
- (k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice (HIGH)