

**University of Washington**  
**Computer Science & Engineering 403: Software Engineering, Winter 2008**  
**Course Syllabus**

**Instructor:**

**name:** Marty Stepp  
**email (at cs):** stepp  
**office:** CSE 466  
**office phone:** (206) 685-2181  
**office hours:** *see course web site*

**TAs:**

Jeremy Brudvik      Felix Chu      Daniel Otero  
jtbrudvi              felixctc      oterod

**Course Content and Learning Objectives:**

This course is an introduction to software engineering using a large group project. You will learn about:

- the phases of software development (the software "lifecycle")
- software requirements analysis and specification
- design concepts and techniques, including Unified Modeling Language (UML)
- implementation practices such as design patterns
- testing, verification, and quality assurance (QA) techniques, including unit testing
- software and tools for software engineering and rapid development
- team and management skills for completing a project in a large group

**Lecture Times:**

MWF 2:30pm - 3:20pm, EE 045  
Th 2:30pm - 3:20pm, EE 045 (discussion section led by TAs)

**Textbooks and Readings:**

There is no textbook, but there will be reading assignments throughout the quarter that will be posted to the course web site to print or read online. You must turn in a typed summary of each reading, either electronically the night before the due date or hard-copy (on paper) in class on the due date. Summaries will not be accepted late. Summary grading guidelines and an example summary can be found on the course web site.

**Course Web Site:**

<http://www.cs.washington.edu/403/>

**Grading:**

Graded work will receive categorized point values, with the following categories and their respective weights:

- 10% individual written summaries of assigned readings
- 10% individual assignments and in-class work
- 60% group project (to be distributed among several phases, as described later in the course)
- 20% final exam: **Tuesday, March 18, 2008, 2:30pm - 4:20pm, EE 045**

Your percentage is mapped onto the 4.0 grade scale roughly as follows. You are guaranteed at least the grade shown below for the percentage shown.

|                   |                   |                   |
|-------------------|-------------------|-------------------|
| 90%: at least 3.5 | 85%: at least 3.0 | 80%: at least 2.5 |
| 75%: at least 2.0 | 70%: at least 1.5 | 60%: at least 0.7 |

## **Turn-in and Lateness Policy:**

Assignments will have due times written on their specifications. Written assignments are submitted by hand in class, at the beginning of lecture on the due date. Programming assignments are submitted electronically.

Project phases are turned in using an online submission system. The URL for this system can be found on the course web site. Assignments will not be accepted by other turn-in methods unless prior permission is given by the instructor or TA. It is your responsibility to ensure that your turn-in is completed successfully.

Summaries of readings are not accepted late. Other assignments may be turned in up to 24 hours late with a 10% penalty, or 24 to 48 hours late with a 20% penalty. No assignments will be accepted more than 48 hours late for any reason. If the assignment is written, it can be slid under the instructor's door to submit it late.

## **Final Exam:**

Exams are open-book and open-notes; you may use your textbook, handouts, printed solutions to your assignments, or any other written materials. No computing devices or others' work or ideas may be used.

If you must miss an exam, you must notify the instructor at least 48 hours in advance of the time of the exam, and you must have a valid excuse. Make-up exams will not be given without instructor's permission.

## **Computing and Labs:**

The course will be taught using Java and its associated tools. If you and your group wish to use a different set of language(s) and tools, you must demonstrate clearly to the instructor that corresponding features and utilities exist in your environment (such as tools for unit testing, source control, development environments, etc.)

If you work from home, it is your responsibility to ensure that your program will run on the school's machines, since that is the environment in which your code will be tested and graded. This also means that your code must constrain itself to language features that exist in the labs.

## **Groups:**

A large portion of the grade for this course will come from working on a large software project in a group with other students. As a group member, you will be expected to do all of the following:

- contribute a significant amount to the analysis, design, implementation, and testing of your project
- meet at least once weekly with your group at a scheduled time
- meet at various times with the instructor and/or TA, together with your group, at a scheduled time
- read and respond regularly to email from your group partners
- communicate with your group partners as needed by email, in person, by phone, or otherwise
- send a weekly group progress email to the instructor
- hold your group partners accountable for their work, and report to the instructor if they fail to do it

## **Collaboration Policy:**

Academic integrity is generally not a large problem in this course, but just to be sure, we want to make a few rules clear. Reading summaries and individual assignments are to be completed by yourself. You may discuss ideas about these assignments with other students. But you should not divulge answers or program code to other students for any reason, including making sure that your answers cannot easily be copied by others.

Much of the work in this course, particularly the large project, is performed in groups. The rules here are similar: Groups may discuss ideas about their design or their project with other groups, but they should not share their actual design documents, code, or other work with other groups.

When in doubt, ask the instructor whether a particular behavior is acceptable under this policy.