

University of Washington, Tacoma

TCSS 360 (Software Development and Quality Assurance Techniques), Spring 2005

Handout 1: Course Syllabus

Contact information:

name: Marty Stepp
email: stepp AT u washington edu
office: Cherry Parks 225
office hours: MW 5:45pm - 6:30pm; TTh 12:50pm - 1:30pm; and whenever door is open, or by appointment
office phone: (253) 692-4540
cell phone: (206) 949-0504

Lecture and lab times:

Section A: MW, 6:45pm - 9:00pm, CP 105
Section B: TTh, 10:30am - 12:45pm, PNK 104

Course content and learning objectives:

This course is an introduction to software development and engineering. In this course, you will learn:

- the phases of software development (the software "lifecycle")
- methods for software analysis and requirements analysis
- several software design concepts and techniques
- the Unified Modeling Language (UML) for software design
- intelligent program implementation practices, such as design patterns
- software verification, testing and quality assurance (QA) techniques, including unit testing with JUnit
- team programming and software project skills for completing a large group project

Textbook:

There is no textbook for this course. There will be required reading assignments posted to the course web site, which students must print or read online.

Course web site:

<http://faculty.washington.edu/stepp/360/>

The course web site is an important part of the class. Lecture notes, handouts, assignments, code files, links, and important announcements will appear on this web site. You are expected to check the course web site at least once per day to see any important announcements that may be posted. You may also post to the class newsgroup, reachable from the main course web site.

Grading:

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

- 15% individual written summaries of assigned readings
- 65% group project
- 20% final exam

Section A: Mon 6/6/2005, 6:45pm - 9:00pm, CP 105

Section B: Thu 6/9/2005, 10:30am - 12:45pm, PNK 104

Grades will be posted regularly to the course web site. Grades may be curved upward at instructor's discretion, but will not be adjusted downward. Your final course grade percentage will be translated into a grade on the 4.0 numbering system roughly by the UW mapping found at the following web site:

<http://www.washington.edu/faculty/facsenate/handbook/04-03-11.html>

If you discover a math error in your grade for a particular assignment, exam, or other work, you may bring this to the instructor at any time during the semester to have it corrected. However, if you believe your grading was incorrect for other reasons, such as overly harsh point deductions or misinterpretation by the grader, you must make this correction within 1 week of the time you get the assignment back. After this week has passed, the instructor reserves the right to refuse to re-examine any incorrectly graded work.

If you feel that the instructor, classmates, or anyone else is discriminating against you in this class, please notify the instructor or a department academic advisor. It is our goal that every student is treated with equal respect in this class.

Written Summaries of Assigned Readings:

At regular intervals, you will be assigned to read articles and excerpts related to the course material. You must turn in a hard-copy (on paper) written summary of each reading, in class on the due date. Summaries will not be accepted late. The summary should be typed. It should occupy no more than one page (expected length: approximately 1/2 page). Assume that the reader of your summary is a competent computer scientist who has *not* read the article before.

Each summary should contain the following:

- a topic sentence that states the name and author of the reading
- at least three additional sentences containing the main idea and several important supporting points and ideas from the reading
- at least one sentence about how this material relates to what has been covered so far in class
- at least two sentences of your own analysis of the paper (Does the author present valid points? Do you agree with the paper? What criticisms do you have, or what changes would you make?)

The summaries will be graded on the following four-point scale:

- 4:** All main concepts and skills mastered and all major questions answered, with only minor errors.
- 3:** Important points made, but contains significant omissions or errors.
- 2:** Substantial missing concepts or errors.
- 1:** Effort shown, but not a significant amount of relevant or correct content.
- 0:** Not turned in, or almost no effort or understanding demonstrated.

An example summary can be found on the course web site, to get a better idea of the format to use.

Turn-in and late homework policy:

Assignments will have due dates and times written clearly on their writeup sheets. Written assignments are submitted by hand in class, at the beginning of lecture on the due date. Programming assignments are submitted electronically, usually before midnight on their due date.

Programming assignments must be turned in using the Catalyst online e-submit system. The URL for this system can be found on the course web site, in the Homework section. Assignments will not be accepted by email, FTP, instant message, or other various turn-in methods unless prior permission has been given by the instructor. It is your responsibility to ensure that your turn-in is completed successfully and on time.

Summaries of readings may not be turned in late. Other assignments may be turned in up to 24 hours late with a 10% penalty, or 24 to 48 hours late with a 25% penalty. No assignments will be accepted more than 48 hours late for any reason. If the assignment is written, it should be slid under the instructor's door to submit it late.

Exams:

There will not be any mid-term examinations in this course, but there will be one final exam. The final exam will take place in class, in the regular classroom. 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 other students' work or ideas may be used.

If you cannot attend lecture on the day of 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 for missing the lecture. Validity of excuses is at the discretion of the instructor and must be verified before the exam can be missed. Failure to attend an exam without reaching agreement with the instructor to miss it results in a grade of zero. Make-up exams will not be given without special permission from the instructor.

Computing and labs:

The on-campus labs such as SCI104, SCI106 and DOU110 provide the recommended Eclipse editor for writing your programming assignments. Eclipse is also available for download at <http://www.eclipse.org/>.

In order to access the school's labs, you will need an active computing account, lab password, and a lab access card. While working in the lab, please obey the lab guidelines outlined at <http://css.tacoma.washington.edu/~lab/> and <http://www.washington.edu/computing/rules/>.

You may also wish to work from home. However, if you do so, it is your responsibility to ensure that your program will run on the school's Windows 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 features of Java that exist in the version of Eclipse used in the lab.

If a problem arises in the lab, such as a broken computer or a service outage, please send email to csslab@u.washington.edu describing the problem. In general, no extensions will be given for lab power or network connection outages, inclement weather, or other problems that occur while an assignment is out, unless that outage spans more than 24 hours or otherwise seriously hinders the ability to complete the assignment as decided by the instructor. The instructor has final say on such matters and will announce any such extensions promptly if granted.

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 least once weekly with the instructor, 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

The group process is not negotiable; no student may complete the project alone. If you cannot or will not perform the above, you will not meet the requirements for this course.

Campus support:

If you would like to request academic accommodations due to a temporary or permanent disability, contact Lisa Tice, Manager for Disability Support Services (DSS) in the Mattress Factory Bldg, Suite 206. An appointment can be made through the front desk of Student Affairs (692-4400), through Student Services (692-4501), by phoning Lisa directly at 692-4493 (voice) or 692-4413 (TTY), or by e-mail ltice@u.washington.edu. Appropriate accommodations are arranged after you have conferred with the DSS Manager and presented the required documentation of your disability to DSS.

Collaboration policy:

Reading summaries and individual homework 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.

You are responsible for making sure that your individual assignment answers cannot easily be seen by other students. If another student submits individual work that is copied from yours, even without your knowledge, both of you will be considered equally guilty. So, for example, you should be very careful not to leave your computer logged in or leave printouts of your answers sitting in the lab printers. The contents of the class newsgroup and your email messages to other students also fall under this policy; so do not post answers or solution code in these areas.

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 divulge their actual design documents or project code to other groups for any reason.

When in doubt, ask the instructor whether a particular behavior violates the spirit and/or intent of the academic integrity policy. Violation of this policy will be punished by reduction of points for all parties involved, and possible referral to the department for further disciplinary action and/or marks on your permanent record.

Lecture Calendar:

The following schedule roughly outlines the quarter. It is subject to change.

Week	M	Tu	W	Th
------	---	----	---	----

March

1	28	29	30	31
	Syllabus, Therac-25 Read: Requirements		Requirements Read: Use cases	

April

2	4	5	6	7
	Use cases Read: Paper prot		Paper Prototyping Read: UML class d	
3	11	12	13	14
	UML class diagrams		UML class diagrams 2 Due: Project SRS Read: UML seq d	
4	18	19	20	21
	UML sequence diag Read: Design Heur		OO Design Heuristics	
5	25	26	27	28
	Present SDS in class Due: Project SDS Read: Prog contract		Progr by contract Read: Conventions	

May

6	2	3	4	5
	Coding conventions Read: Patterns 1		Singleton, Flyweight Read: Patterns 2	
7	9	10	11	12
	Factory, Prototype Read: Patterns 3		Strategy, State Read: Patterns 4	
8	16	17	18	19
	Due: Code Milestone Composite, Decorator, Command, Adapter Read: Pair/extreme		Pair programming Read: Testing	
9	23	24	25	26
	Testing Read: Unit testing		Unit testing Read: Profiling	
10	30	31		
	Profiling			

June

			1	2
			Present project in class Due: Final Code and Test Cases	
11	6	7	8	9
	Final exam			Final exam