

CS 4320 SOFTWARE TESTING AND QUALITY ASSURANCE: SYLLABUS (SPRING 2008)

Contact:

Instructor : Ted Billard
Email : ted.billard@csueastbay.edu
Office : North Science 216
Phone : 510-885-3437
Instructor Home Page : <http://www.mcs.csueastbay.edu/~billard>
Course Home Page : <http://www.mcs.csueastbay.edu/~billard/se/cs4320.html>

This course will be offered online using the Blackboard system, with the exception of the mid-term and finals exams (see below).

Catalog Description:

Concepts and issues in the testing and quality control of large software projects. Topics include white box, black box, unit, integration, and validation testing; quality assurance through planning, review, and use of software metrics.

Summary:

The course covers concepts and issues in the testing and quality control of software projects. The main points are white box, black box, unit, integration, and validation testing. Other topics include code correctness, coverage, cyclometric complexity, boundary-value analysis, code walkthroughs, JUnit test, and metrics. The goal is to prepare students for testing in the real world of software engineering.

Learning Outcomes: The successful student will be able to

- methodically create and execute a test plan,
- understand the fundamentals of white and black box test,
- analyze the details of code sources for faults.

Prerequisite: CS 3240 Data Structures and Algorithms. CS 4320 is not a programming course but students should know how to program in C/C++.

Text/Exercise Packet: <http://www.mcs.csueastbay.edu/~billard/se/cs4320.zip>

Grading:

Exercises : 40% [due on Sunday night of specified week, **no late submissions**]
Mid-Term Exam : 25% [Sat. May 3 12:30-2:30pm]
Final Exam : 35% [Sat. June 7 12:30-2:30pm]

Students must be able to attend these exam dates. The exams are held **in-class** (room: TBD). All students must bring a Photo ID.

Instructor Communication:

Precedence	Mode	Frequency
1.	Email	7 days/week at least AM and PM
2.	Phone	We can set up calls
3.	Office Hours	Sat. morning, confirm date/time in advance
4.	Optional Labs	In conjunction with office hours
5.	Other In-person Advising	By appointment
6.	Chat Room	As needed
7.	Virtual Classroom	As needed

Exercise Modules:

#	Due	Wt	Topics	Goals
1.	Wk 1		Proof of Correctness	Prove code is correct mathematically using invariants (mult.c)
1.	Wk 2	5%	Complete assignment	
2.	Wk 3	4%	Code Walkthrough	Desk check code for correct results
3.	Wk 3	3%	Code Coverage	Introduce statement, decision, condition coverage
4.	Wk 4		White Box Test	Cyclometric complexity, flowgraphs, paths (minmax.c)
4.	Wk 5	7%	Complete assignment	
5.	Wk 6		Black Box Test	Validation test using boundary-value analysis (BVA.txt)
5.	Wk 7	7%	Complete assignment	
6.	Wk 7	2%	Source Control	Structure software versions using SCCS
7.	Wk 8	2%	Unit Test	Black box test a Btree function
8.	Opt	0%	Metrics	Examine software measurements
9.	Opt	0%	Coverage Analyzer	Write an automated test script to improve code coverage
10.	Wk 8	3%	All Pairs Test	Automated test using all pairs instead of all combinations
11.	Wk 9	3%	JUnit Test	Test data structures using JUnit test harness
12.	Wk 10	4%	Integration Test	JUnit test using drivers and stubs
13.	Opt	0%	General Concepts	Study life cycle, quality assurance, metrics, code verification
		40%		

Collaboration/Copying: Unless otherwise stated, do your own work in this class. Students may ask help from other students but only at the conceptual (lecture notes) level. Examinations are strictly individual. Violations will be prosecuted to the full extent of the University rules. At the final exam, all students will be asked to sign a form indicating their academic honesty.