Syllabus
CS 525 Software Testing and Quality Assurance
Spring 2008

Lectures  W 06:20 PM - 08:50 PM, HT 212
Instructor  Dr. G. Dimitoglou (E-mail: dimitoglou@hood.edu, Office: HT 261)
Office Hours: Mon-Wed-Thu 5:00-6:00pm or by appointment.

Course Description
This course examines the theory and practice behind software testing and quality assurance. Emphasis is placed on understanding the software testing process, planning, strategy, criteria, and testing methods, as well as software quality assurance concepts & control process. Topics will include test models, test design techniques (black box and white-box testing), integration, regression, measurement, unit testing, slicing and debugging, inspection, and software metrics. Emerging concepts and their impact on testing will also be examined. This is both a theoretical and hands-on course. Multiple software testing suites will be used during the semester to enforce student mastery of the material.
Prerequisite: CS 524 or permission of the instructor.

Course Objectives
By the end of the course students will be able to:
(a) Apply appropriate software testing techniques to different development projects
(b) Perform software verification and validation activities and provide rationale for selecting and combining them within a software development process.
(c) Test software programs against software requirements using a test plan.
(d) Apply software metrics to development projects and perform software cost estimation.
(e) Write test plans and procedures.
(f) Explain the concepts of correctness and completeness as they relate to software quality

Text(s)
(b) Lecture notes and scholarly papers will be provided via Blackboard (see Reading List section).

Assessment
Mid-Term (30%), Final Exam (35%), Homework (35%)

Policies-Guidelines
1. Adhering to the Academic Honesty Policy/Honor Code is student responsibility. Deviation from the policy will not be tolerated. Discussions with classmates are permitted but deliverables must be your own, individual work.
2. Assignments are due in the beginning of class. In fairness to those students who submit their work on time, late deliverables will receive zero points. No exceptions.
3. You are responsible for the content of reading assignments, lectures, handouts, announcements and schedule changes made in class whether or not you are present. If you must miss a class, be sure to check Blackboard.
4. Attendance is expected at each class meeting. While there is no attendance grade, it is in your own best interest to attend class, as your grade will almost certainly suffer indirectly if you choose not to attend.

5. The material in the course is, inherently, cumulative. Be aware, if you fall behind, it may be difficult to catch up.

**Topics & Schedule (tentative)**

NOTE: The examination dates (March 19, May 7) are firm. Mark your calendars. However, the order and coverage of topics is tentative.

<table>
<thead>
<tr>
<th>Session</th>
<th>Date</th>
<th>Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Jan-30</td>
<td>Introduction</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Overview: Software Testing and Analysis</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A Test &amp; Analysis Framework</td>
</tr>
<tr>
<td>2</td>
<td>Feb-06</td>
<td>Software Testing Principles</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Testing within the Software Development Process</td>
</tr>
<tr>
<td>3</td>
<td>Feb-13</td>
<td>Techniques: Finite Models</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Test Case Selection &amp; Adequacy</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Functional Testing</td>
</tr>
<tr>
<td>4</td>
<td>Feb-20</td>
<td>Combinatorial Testing</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Structural Testing</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Data Flow Testing</td>
</tr>
<tr>
<td>5</td>
<td>Feb-27</td>
<td>Model-Based Testing</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OO Testing</td>
</tr>
<tr>
<td>6</td>
<td>Mar-05</td>
<td>Fault-based Testing</td>
</tr>
<tr>
<td>7</td>
<td>Mar-12</td>
<td><strong>No Class-Spring Break</strong></td>
</tr>
<tr>
<td>8</td>
<td>Mar-19</td>
<td><strong>Midterm Examination</strong></td>
</tr>
<tr>
<td>9</td>
<td>Mar-26</td>
<td>Test Execution</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Inspection</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Program Analysis</td>
</tr>
<tr>
<td>10</td>
<td>Apr-02</td>
<td>Process Planning and Monitoring</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Integration and Component-based Testing</td>
</tr>
<tr>
<td>11</td>
<td>Apr-09</td>
<td>System, Acceptance, Regression Testing</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Automation Testing</td>
</tr>
<tr>
<td>12</td>
<td>Apr-16</td>
<td>Documentation Analysis and Test</td>
</tr>
<tr>
<td>13</td>
<td>Apr-23</td>
<td>TBA</td>
</tr>
<tr>
<td>14</td>
<td>Apr-30</td>
<td>TBA</td>
</tr>
<tr>
<td>15</td>
<td>May-07</td>
<td><strong>Final Examination</strong></td>
</tr>
</tbody>
</table>
Exams
The exams consist of expository and problem-solving questions similar to the homework assignments. The exams are closed book. The exams do not require memorization, only mastery of the material. Their content is cumulative, i.e. they address the material covered up to the day of the exam. If a student misses the midterm exam due to an emergency (as agreed in advance by the instructor and/or by providing definite proof of medical or legal reason), there will be no makeup exam: the final will become proportionally more important. If a student misses the midterm exam without prior agreement by the instructor and/or definite proof as to the medical or legal reasons, the student receives a zero grade for the exam. The final exam is mandatory.

Homework Guidelines

Programming Assignments
(1) Platform: You may use any programming language/compiler available on PLUTO or WYRD. If the assignment can not be compiled and executed on any of these systems, it will receive zero points.

(2) Submission: Electronic, via the digital drop-box on Blackboard. Do not email your assignment (it will be discarded). Every submission must include: (a) the Source code, (b) a README file to: briefly describe what the code does; what the code does not do (with respect to the assignment); how to compile (e.g. compiler version and flags); how to execute the source code, (c) a sample input file (if applicable), (d) a sample output file or, execution output capture/screenshot Submissions should not include executables, the original assignment or files irrelevant to the course.

Every submission must contain all of the elements above in a single TAR or ZIP file that is named following the convention: hw<n>_lastname.tar. So, if I were to submit an assignment for Homework #3, the file would be named: hw3_dimitoglou.tar

Non-Programming assignments
(1) Should be typed, unless otherwise specified in the assignment.

Programming Assignment Grading Rubric
Code does not compile or execute on designated system (PLUTO/ WYRD) Earns 0%
Code has been developed but does not compile. Earns no more than 5%
(Code must be relevant to the assignment, not a “Hello World” submission.)
Code runs but with run-time failures, produces erroneous results. Earns no more than 20%
(Code must be relevant to the assignment, not a “Hello World” submission.)
Code runs only with instrumented or fixed input. Earns no more than 30%
Code runs but partially addresses assignment requirements. Earns no more than 50%
(Code must be relevant to the assignment, not a “Hello World” submission.)
Code runs and addresses all assignment requirements but is not optimal. Earns no more than 75%
(Optimality determined based on asymptotic complexity. If there is a faster algorithm or implementation then it is not considered optimal.)
Code runs and addresses all assignment requirements and is optimal. Earns no more than 90%
Assignment complies fully with the homework submission guidelines. Earns a full 10%
(e.g. adequate documentation, proper file naming, correct submission procedure etc)
Failure to comply with any one of the homework guidelines. Deducts a full 10%
(e.g. poor documentation, failure to adhere to file naming, submission guidelines etc)
Grading Scale(s)

<table>
<thead>
<tr>
<th></th>
<th>Undergraduate Scale</th>
<th>Graduate Scale</th>
</tr>
</thead>
<tbody>
<tr>
<td>≥ 93.3%</td>
<td>A</td>
<td>≥ 93.3%</td>
</tr>
<tr>
<td>≥ 90%</td>
<td>A-</td>
<td>≥ 90%</td>
</tr>
<tr>
<td>≥ 86.6%</td>
<td>B+</td>
<td>≥ 86.6%</td>
</tr>
<tr>
<td>≥ 83.3%</td>
<td>B</td>
<td>≥ 83.3%</td>
</tr>
<tr>
<td>≥ 80%</td>
<td>B-</td>
<td>≥ 80%</td>
</tr>
<tr>
<td>≥ 76.6%</td>
<td>C+</td>
<td>≥ 76.6%</td>
</tr>
<tr>
<td>≥ 73.3%</td>
<td>C</td>
<td>≥ 73.3%</td>
</tr>
<tr>
<td>≥ 70%</td>
<td>C-</td>
<td>≥ 70%</td>
</tr>
<tr>
<td>≥ 66.6%</td>
<td>D+</td>
<td>≤ 69%</td>
</tr>
<tr>
<td>≥ 63.3%</td>
<td>D</td>
<td></td>
</tr>
<tr>
<td>≥ 60%</td>
<td>D-</td>
<td></td>
</tr>
<tr>
<td>≤ 59%</td>
<td>F</td>
<td></td>
</tr>
</tbody>
</table>

Reading List (by topic area)
A number of the following papers will be provided electronically (on Blackboard). They will either be referred to during the lectures or, will appear as Reading Assignments throughout the term.

GENERAL/INTRODUCTION

CODE INSPECTION

TEST COVERAGE & ADEQUACY

ASSERTIONS

ERROR SEEDING

COVERAGE CRITERIA
DEPENDENCY ANALYSIS.PROGRAM SLICING

FAULT BASED TESTING

REGRESSION TESTING

OBJECT ORIENTED TESTING

INTERPROCEDURAL ANALYSIS

SOFTWARE VERIFICATION

DATA FLOW ANALYSIS

SPECIFICATION BASED TESTING AND ANALYSIS

CONCURRENCY ANALYSIS

EXPERIMENTAL STUDIES

SOFTWARE SAFETY