Computer Science 521/621
Advanced Software Engineering—Analysis and Testing
Fall 2011
Instructor: Prof. Leon J. Osterweil (ljo@cs.umass.edu)
Office: CS 302

Class meeting times: T, Th 2:30-3:45
Meeting Room: CS 140
Course Web Site: http://laser.cs.umass.edu/courses/cs521-621.Fall 2011/

Teaching Assistant: Wenzhe Wu (wenzhe@cs.umass.edu)
Teaching Assistant Office Hours: T, Th 11-12

Text: There is no assigned text for this course. But readings from the research literature will be assigned. The readings will be made available through the course web site.

Supplementary References: The following two books are sources of good information and could be sources of useful background material.
• Ghezzi, Jazayeri, and Mandrioli, Fundamentals of Software Engineering (Prentice-Hall)
• Pezze and Young, Software Testing and Analysis: Process, Principles, and Techniques (John Wiley)

Instructor slides: Will be on course web site (hopefully, usually) before lecture

Instructor Office Hours: Tuesdays and Thursdays 4:00-5:00, and by appointment

Course Summary: The course will address a range of approaches to the analysis and testing of software artifacts. Emphasis will be placed upon the strengths and weaknesses of various approaches and understandings of the ways in which the complement each other. Primary emphasis will be placed upon testing and analysis for functionality of executable code. But the course will also address testing and analysis for such other software characteristics as speed, robustness, and security, and the applicability of all of these approaches to pre-code software artifacts.

Educational Objectives: The course will aim to instill in students a deep understanding of the nature of quality in software and of various approaches to determining whether quality has been achieved. Much of the course will be aimed at building understandings of the nature and effectiveness of various approaches to software quality determination. The characteristics of a limited number of specific implementations of some of these approaches (e.g. in software tools and evaluation systems) will also be presented more as illustrations than as a survey of current tool offerings.

Evaluation: Approximately six assignments will be made during the course. In addition there will be a mid-term examination as well as a final examination. The student’s course
grade will be based primarily upon grades received on the examinations and the assignments, although some weight will be given to class participation and the instructor’s informal evaluation.

**Examinations:** There will be a mid-term examination and a final examination. Both examinations will be in-class, closed book, closed notes examinations.

**Grading:** The final examination will be weighted as the single most important factor in determining the student’s final course grade. But the sum of the evaluations of all of the assignments will be comparable (and perhaps greater) in weight in determining the student’s final course grade. The weight of the mid-term examination will be greater than one assignment, and significantly less than the weight of the final examination.

**621 Students (only):** Students enrolled in 621 will be asked to do more work and to display deeper insights in the responses to the assignments and to the questions on the mid-term and final examinations. Usually this additional work and depth will be requested in the form of additional questions, or further elaborations of questions.

**Course Policies:** Students are expected to attend class. The official means of communication for this course will be in-class announcements. Missing class is not an excuse for failure to act as required by these announcements. In addition, announcements related to issues that arise between class meetings will be made electronically. In particular, clarifications of assignments, changes to due dates, etc. may be reported to you via postings to the course web site and/or electronic mail. It is the student’s responsibility to log in and check the course web site, and your mail, regularly.

Copies of handouts, homework assignments, and course notes will be posted for downloading from the class website. The home page for this course is [http://laser.cs.umass.edu/courses/cs521-621.Fall11/](http://laser.cs.umass.edu/courses/cs521-621.Fall11/)

**IMPORTANT:**
The work submitted for grading must be your own work. Submission of work that is not your own is considered academic dishonesty. Computer Science department policy specifies that the penalty for academic dishonesty is 1) a final course grade of `F', and 2) possible referral to the Academic Dishonesty Committee.

You may be using copyright-protected software in this course. United States Federal law and license agreements between the University and various software producers prohibit copying this software for any purpose. Such activity will be regarded as a form of academic dishonesty and will be dealt with as such.

An incomplete will usually be given only when documented, exceptional circumstances beyond your control have made it impossible to complete the assigned work before the end of the semester. It is your responsibility to contact the professor regarding any such problems as those circumstances develop. Note that the general rules of the University allow an incomplete only if most of the work has been completed before the end of the
semester so that the incomplete can be finished within the first four weeks of the immediately following semester.

Professor Osterweil will be available immediately after class and during posted office hours. If you cannot arrange for a meeting during those hours, please make an appointment via e-mail.

**Shortened Outline of Course Topics:**

1. Introductions to
   a. This course
   b. Software Engineering
   c. Testing and Analysis
2. Foundations
   a. Discrete Mathematics
   b. Graph Theory
   c. Terminology and notation
3. Software Inspections
4. Assertions
5. Overview of Dynamic analysis
6. Test Coverage Approaches and Measures of Testedness
   a. Dataflow testing
   b. Fault Based testing
   c. Regression Testing
   d. Error Seeding
7. Mutation Analysis
8. Testing for non-functional properties
9. Fault Localization
10. Symbolic Execution
11. Overview of Static Analysis
12. Data Flow Frameworks and Analysis
13. State Propagation
14. Concurrency Analysis
15. Resource Analysis and Scheduling
16. Finite State Verification and Model Checking
17. Analysis of non-functional properties
   a. Robustness
   b. Speed
   c. Safety
   d. Security
18. Lifecycle Testing and Analysis—Building Quality In
   a. Software Development Lifecycle Approaches
      i. Waterfall
      ii. Extreme Programming
      iii. Scrum
   b. Integrated Lifecycle Testing and Analysis
19. Evolutionary Development