CS 5391 Survey of Software Engineering
Spring 2006

Course Instructor:
Name: Mark Grechanik
Office: Nueces 255
Email: mg43@cs.txstate.edu
Web Page: http://www.cs.utexas.edu/users/gmark
Office Hours: W 5:00 – 6:00 PM

Class Meets:
S 9:00 – 11:45 AM in classroom RRHEC 366 and via ITV in San Marcos

Prerequisites:
Basic understanding of the software development steps: compilation, linking, debugging, etc.
Students are expected to have taken CS Data Structures with a grade of C or better or their equivalents
Familiarity with object-oriented programming, specifically C++ and/or Java
If you have not met these prerequisites, you must see me.

Required Textbook(s):

There is one mandatory textbook for this course: Frederick P. Brooks Jr., The Mythical Man-Month, Addison-Wesley, 2nd edition, August 1995. I expect all students to read this book in the first two weeks of this course in parallel with other reading assignments. You will be able to integrate your class notes with copies of slides that I use in class and pointers to papers and web sites relevant to the material discussed in class, all of which I will post on the class web site.

Optional Textbooks:

Carlo Ghezzi, Mehdi Jazayeri, and Dino Mandrioli, Fundamentals of Software Engineering, Prentice Hall, 2nd edition, September 2002.

Robert L. Glass, Facts and Fallacies of Software Engineering, Addison-Wesley, October 2002.

Roger S. Pressman, Software Engineering: A Practitioner's Approach, (McGraw-Hill Series in Computer Science) McGraw Hill, 5th edition, June 2000.

Shari L. Pfleeger, Software Engineering: Theory and Practice, Prentice Hall, 2nd edition, February 2001.

Course Content:

The goal of this course is to provide a broad but solid overview of software engineering. Software engineering is a discipline of designing, implementing, and maintaining software systems. Even though it is the youngest among other engineering disciplines, its importance cannot be overestimated. Since software has become ubiquitous in today's world it affects all aspects of our lives. However, building software requires significant resources and knowledge, and often ends as a failure. In this course we review different areas of software engineering with the concentration on practical aspects of building and maintaining large-scale systems. Lectures will be based mostly on research papers written by accomplished software engineers and computer scientists, and present cutting edge results and the state-of-the-art of major areas of software engineering. Students will be introduced to various topics of software engineering and different domains to which methods and tools learned with each topic are applicable.

Course Goals:

The course will consist of readings, two exams, and a final project. The goals of the course are to:

- Understand the sources of the complexity of large-scale software systems;
- Deal with appropriate abstractions;
- Build and maintain software, and to
- Integrate various tools and methods with existing system development processes.

Course Topics:

- Fundamentals, Processes, and Requirements
- Evolution and Maintenance
- Software Architecture
- Software Design
- Programming models
- Procedural and Structured
- Object-Oriented
- Aspect-Oriented (AOP)
- Feature-Oriented (FOP)
- Adaptive (AP)
- Intentional (IP)
- Component-Based Software Engineering
- Generic model of component infrastructures
- COM/COM+/.Net
- CORBA
- EJB
- Performance considerations

Grading:

- Quizzes: 10%
- Midterm: 25%
- Final: 40%
- Project: 25%

Method of evaluation:

The instructor uses a relative grading method such as grading on the curve.

Blackboard:

All course materials will be made available to you via the university BlackBoard. It is YOUR responsibility to set up your account with the Blackboard and use it to read course announcements and participate in discussion groups. If you have a problem using the Blackboard, then please use the help desk to resolve all your problems.

Attendance policy:

Attendance of lectures is not required. Students are responsible for contacting the instructor for clarification of assignments or additional help as needed. The instructor is always available via e-mail and will be available for office hours as arranged.

Special Accomodations:

Students having special needs/disabilities which require accomodations for successful completion of this course must notify the Office of Disability Services (or course instructor who will notify Office of Disability Services) as soon as possible. Failure to do so in a timely manner may result in accomodations not being made as necessary.

General:

- We meet on Sat @ 9:00PM in Avery 366 +ITV
- We take a 10 min brake @ 10:10AM
- Some lectures will start with a 5-10 min quiz
- Lecture attendance is not required
- LAST DAY to drop the class with a W grade: Feb 3
- Spring break: March 12-19
- LAST DAY to Drop the course: April 20@ 5PM
- LAST CLASS DAY: April 29
- Exams
- Midterm: March 4 @ 9:00AM
- Final: Saturday, May 6 @ 9:30AM
- You must attend the midterm and final exams and complete your project to satisfy the baseline requirements set by me to be considered for a grade of B or better.

- No makeup exams and quizzes will be permitted.
- The course project submission date is Sunday, May 7, 2006 at 10:00PM via email gmark@austin.rr.com. You shall receive a confirmation receipt from me. Failure to submit your project by the deadline shall result in loosing your project grade

Cooperation policy:

I encourage you to discuss the problem sets and programming assignments with your colleagues. I welcome discussions of possible interpretations of questions, solution approaches, and relevant technical details. You are also welcome to use existing public libraries in your programming assignments. Such activities qualify under approved collaboration practices and you are welcome to take advantage of them.

Note that cooperation is not the same thing as cheating. It is OK to ask someone about the concepts needed to do homework or project assignments. However, ***copying*** other people's code or solution sets is strictly prohibited. The quizzes, project assignments, and exams must be the work of students turning them in. Students who violate University rules on scholastic dishonesty are subject to disciplinary penalties, including the possibility of failure in the course and/or dismissal from the University. Because such dishonesty harms the individual, all students, and the integrity of the University, policies on scholastic dishonesty will be strictly enforced. Acts that exceed the bounds defined by the approved collaboration practices will be considered cheating.