ECE 417/617 Elements of Software Engineering
Spring 2007

Instructor: Stan Birchfield
Office: 207-A Riggs Hall, 656-5912, stb at Clemson
Office Hours: 3:30-4:30 MWF, or by appointment
Course website: **http://www.ces.clemson.edu/~stb/ece417**
Text (recommended):

- Bjarne Stroustrup, *The C++ Programming Language*, Addison-Wesley, 2000.
- David Gustafson, *Schaum's Outline of Software Engineering*, McGraw-Hill, 2002
- Roger S. Pressman, Software Engineering, 6th ed., McGraw-Hill, 2005.

Prerequisites: ECE 329, ECE 352, MTHSC 419; C programming, some familiarity with object-oriented programming

Overview: In this course students will learn to build high-quality, reliable, and extensible software systems that are too large for a single programmer. Emphasis is placed upon good programming practices (object-oriented design and clean, well-documented code), teamwork skills (planning, communicating, and interacting), and bringing the project to completion (testing and verification). A semester-long project complements the theory with practice.

Objectives: By the end of the course, students should be able to do the following:

- *Fundamental concepts.* Describe the basic concepts and terms of software engineering, including requirements elicitation, tasks and activities, project roles, analysis and system modeling, object design, user interface design, testing, management, life cycles, the twelve XP practices, mythical man month, and open source. Draw and analyze simple UML diagrams. Describe object oriented programming concepts and calculate the functionality of code containing constructors, destructors, copy constructors, and assignment operators.
- *Tools.* Check in and update code using a revision control system (CVS). Create a new workspace with an IDE (Visual Studio), navigate between files, and use the debugger. Create and modify a graphical user interface using an existing library (Win32 and MFC).
- *Programming and teamwork skills.* Write clean, well-documented C++ code to achieve desired functionality. Write code that can be read and modified by others. Read and modify code written by others. Work with a team of programmers to build a large software system. Communicate effectively with team members and take initiative to contribute to the overall goal. Test code thoroughly, check in only code that works, and refactor code as necessary.

Topics:	**Lectures**
• introduction	1
• C/C++ programming language and tools	5
• software life cycles	3
• software process and modeling (UML)	3
• project management	3

• agile methods	3
• requirements	3
• system design	3
• testing	3
• risk analysis	3
• user interfaces	3
• formal methods	3
• group meetings	7

TOTAL	43

Grading:

1. *Project Milestones.* There will be approximately seven project milestones during the semester. Each group will receive a grade for their work for each milestone. In general, all the members of a group will receive the same grade, although provision will be made to properly adjust grades for members who perform exceptionally well or who fail to pull their share of the workload.
2. *Quizzes.* There will be six or seven short quizzes during class, which may be announced or unannounced. A student who keeps up with lectures and reading assignments should do well on these quizzes. Unless agreed upon in advance by the instructor, missed quizzes will receive a grade of zero. To receive a passing grade in the course, a student must take at least four quizzes. The grade of the lowest quiz may be dropped, subject to instructor discretion.
3. *Individual Assignments.* There will be one individual programming assignment. Late assignments will be accepted at a penalty of 10 points per day (according to a six-day work week), up to a maximum of 35 penalty points; assignments turned in more than one week late will receive a zero. In addition, students will research a topic of their choosing within the scope of the course, presenting their findings in a written report and oral presentation. Students must turn in their own work. As always, students who violate University rules on academic dishonesty will be subject to disciplinary penalties, such as failure in the course and/or dismissal from the University.
4. *Exam.* There will be a closed-book, closed-notes final examination covering the material for the entire semester. The instructor should be notified of any conflict that would prevent a student from taking the exam at least one week in advance, so that alternative arrangements can be made. Without prior approval, a missed exam cannot be made up except in cases of extreme urgency and importance (e.g., sudden illness, death in the immediate family).
5. *Grading.* Grades will be determined by the following formula: project (40%), individual assignment and paper (20%), quizzes (20%), final exam (20%).