

## ECE 417/617 Elements of Software Engineering Spring 2005

**Instructor:** Stan Birchfield, 207A Riggs Hall, 656-5912, stb at clemson

**Office Hours:** 10:45-11:45 TTh, or by appointment

**Grading assistant:** TBD

**Class meets:** 2:30-3:20 MWF, 227 Riggs Hall

**Website:** <http://www.ces.clemson.edu/~stb/ece417>

**Text:**

- D. H. Steinberg and D. W. Palmer, *[Extreme Software Engineering: A Hands-On Approach](#)*, Pearson Prentice-Hall, 2004.

**Prerequisites:** 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.

**Grading:** project (50%), individual programming assignments (20%), quizzes (30%)

**Academic integrity:** “As members of the Clemson University community, we have inherited Thomas Green Clemson’s vision of this institution as a ‘high seminary of learning.’ Fundamental to this vision is a mutual commitment to truthfulness, honor, and responsibility, without which we cannot earn the trust and respect of others. Furthermore, we recognize that academic dishonesty detracts from the value of a Clemson degree. Therefore, we shall not tolerate lying, cheating, or stealing in any form.” – from the university’s web page on academic integrity:

[http://www.cs.clemson.edu/html/academics/academic\\_integrity\\_2002.html](http://www.cs.clemson.edu/html/academics/academic_integrity_2002.html)