# ECE 417/617 Elements of Software Engineering
## Spring 2012

**Instructor:** Stan Birchfield
**Office:** 209 Riggs Hall, 656-5912, stb at Clemson
**Office Hours:** Wed 4-5, or by appointment
**Lectures:** MW 2:30 – 3:45, Riggs 226
**Course website:** http://www.ces.clemson.edu/~stb/ece417
**Text:**

- Steve McConnell, *Code Complete*, 2nd ed., Microsoft Press, 2004. **(required)**
- Bjarne Stroustrup, *The C++ Programming Language*, Addison-Wesley, 2000.
- Roger S. Pressman, *Software Engineering*, 6th ed., McGraw-Hill, 2005.

**Prerequisites:** ECE 329, ECE 352, MTHSC 419; C programming skills

**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, such as requirements elicitation, tasks and activities, project roles, analysis and system modeling, architecture, object design, user interface design, testing, management, life cycles, agile programming, formal methods, mythical man month, and open source. Draw and analyze 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 | 3 |
| • software tools (CVS, VC++) | 2 |
| • software process | 2 |
| • modeling | 2 |

| | |
|---|---|
| • design | 2 |
| • user interface design | 2 |
| • testing | 2 |
| • project management | 2 |
| • software practice | 2 |
| • formal methods | 2 |
| • meetings / code reviews | 21 |
| | --- |
| TOTAL | 43 |

**Grading:**

1. *Attendance*. Because of the group-based nature of the course, attendance in class and at group meetings is important. Therefore, attendance will be taken at the beginning of each class.
2. *Group Project.* Groups will provide weekly progress updates, along with goals for the upcoming week. In addition, they will produce work products at various intervals, including models, diagrams, schedules, and checked-in source code.
3. *Individual Project.* Students will complete a small individual project produce Groups will provide weekly progress updates, along with goals for the upcoming week. In addition, they will produce work products at various intervals, including models, diagrams, schedules, and checked-in source code.
4. *Written Assignments.* To complement the hands-on experience of the project, students will be expected to learn material on their own regarding the principles and established findings of software engineering. This material will primarily come from the required text, but it may come from other sources as well. This learning process will be measured via written weekly assignments.
5. *Final exam.* There will be a final examination covering the material for the entire semester. 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).

Grades will be determined by the following formula: group project (50%), individual project (30%), written assignments (10%), final exam (10%). Final grade is then reduced by *max(0, (a-2)*5)* points, where *a* is the number of unexcused absences.

Students taking the graduate-level version of the course (ECE 617) must, in addition to the above, write a 2-3 page summary of an *IEEE Transactions of S/W Engineering* paper of their choice.