20EECE180 SOFTWARE METHODS I
SPRING QUARTERS

Required/Elective: Required for EE and CompE majors

Catalog Data: 20EECE180. Software Methods I. Credits 4. Introduction to software methods for solving engineering problems. Emphasis is on the software development process including creating problem requirements, systematic software design, correct and maintainable C++ coding, and efficient software testing methods. Prereq: None.


References: C++ reference guide, Visual Studio documentation, Coding style guide

Coordinator: Hal Carter, Professor of Electrical & Computer Engineering.

Goals: This course is designed to introduce the student to the development of software programs to solve typical engineering problems, particularly for embedded digital systems. Upon successful completion of this course the student will be able to specify, design, implement, and test intermediate-level computer programs written in C++.

Prereqs by Topic: 1. Binary number representation of integers
2. Elementary boolean logic

Topics: 1. Software development methods
2. Object-oriented design and coding
3. Modularity, functionality, recursion, and templates
4. Array, list, and stack data structures
5. Simple algorithms including analysis of their performance
6. Software test planning and application methods

ABET Outcomes: a, c, e, g

Lab Projects: Eight weekly projects consisting of software solutions to practical problems occurring the general, engineering, and embedded systems domains.

Components: Engineering Science: 2 credits (50%), Engineering design: 2 credits (50%)

Prep/Updated: Hal Carter, Ph.D. Date: March 2009
Learning Objectives

By the end of the course each student shall be able to:

1. Apply the engineering method to create software programs to solve engineering problems including embedded computing problems (a, c, e, g)
2. Apply object-oriented design and coding using the C++ language (c)
3. Exploit functional modularity to support “divide and conquer” design and coding (e)
4. Employ array, list, and stack data structures in computer programs (e)
5. Implement simple algorithms and analyze their performance (e)
6. Perform software test planning and apply test methods to assure correctness of computer programs (e)

Legend (ABET course outcomes):

a. an ability to apply knowledge of mathematics, science, and engineering
c. an ability to design a system, component, or process to meet desired needs.
e. an ability to identify, formulate, and solve problems that arise in their discipline and its applications.
g. an ability to communicate effectively.