Object-Oriented Software Engineering

EE 467 -- Fall 2002

Syllabus

Instructor: Prof. Nancy Reed, 439 Holmes Hall, nreed@spectra.eng.hawaii.edu (or nreed@wiliki), (808) 956-9735.
Lectures: MWF 11:30-12:20, POST 214 (Kim Lab).
Office hours: MWF 1:00-2:00, Holmes 439.
Credits: 3.
Textbook: "C++ How to Program" by H.M. Deitel and P.J. Deitel, Prentice Hall, Upper Saddle River, NJ, 2001. ISBN: 0-13-089571-7.

Quick Links: Messages, Important Dates, Course Description (pdf), Exam Information, Lecture Material, Assignments and Projects, Groups, Grading Criteria , Help, Code Examples.

Overview

This course is designed for upper-division electrical engineering students. The course covers the development of large software projects using software engineering and object-oriented programming techniques. During the term, the students will complete several individual programming exercises, team projects, and examinations. The C++ programming language is used.

Prerequisites

Prerequisites include a grade of C or better (or equivalent knowledge) in EE160 (introductory programming in C), EE260 (basic circuit design), and EE367 (data structures in C). A minimum of 1 year of programming experience in C or C++ is expected.

Messages

Read messages frequently here .

Course Requirements

There are three components to your grade in this course, individual programming assignments (25%), team projects (35%), and exams (40%). See the grading criteria and assignment descriptions for more details.

Lectures. You are expected to attend lectures and you are responsible for all information given out in lecture. Available lecture material will be posted here.

Programming assignments and team projects. Programming assignments will be done **individually**. See the assignments page for more details. Programming projects will be done in **groups** of approximately 4 students. For more information see the assignments page.

Exams. There will be two in-class midterm exams and one final exam. There may be pop quizzes during lectures.

Topics Covered

- Review of control structures, functions, and primitive data types
- Object-oriented programming: Object-oriented design; encapsulation and information hiding; separation of behavior and implementation; classes, subclasses, and inheritance; polymorphism; class hierarchies
- Fundamental computing algorithms: simple searching and sorting algorithms (linear and binary search, selection and insertion sort)
- Software engineering and evolution: Software maintenance; characteristics of maintainable software; reengineering; legacy systems; software reuse

The expanded course description can be found here (pdf).

Enjoy!

Back to the top
