EE 205  Object-Oriented Programming

Designation: Required

Catalog Description: EE 205  Object-Oriented Programming(3). Second level programming course for Computer Engineers. Introduces the object-oriented programming paradigm focusing on the definition and use of classes along with fundamentals of object-oriented design in a modern object-oriented language such as C++. Other topics include complex data structures, simple searching and sorting techniques and an introduction to software engineering issues.

Credits: 3

Pre-and Co-requisites: Pre-requisites: EE160 – Programming for Engineers, or instructor consent.

Class/Lab Schedule: 2 lecture hours and 2 lab hours per week.

Topics Covered:
- Programming in C and C++ - control structures, functions, pointers, arrays and structures. (3 hours)
- Introduction to Software Engineering - life cycle: analysis, design, implementation and maintenance.(2 hours)
- Object-Oriented programming model – encapsulation, information hiding, polymorphism, inheritance. (3 hours)
- C++/Object-Oriented concepts – classes, information hiding, encapsulation.(4 hours)
- Simple linked data structures – linked lists, stacks, queues. (3 hours)
- C++/Object-Oriented concepts – operator overloading (3 hours)
- C++/Object-Oriented concepts - inheritance, and polymorphism.( 3 hours)
- C++ - templates, streams and stream I/O. (3 hours)
- C++ - File processing, string class, string stream processing . (3 hours)
- C++ - legacy code topics, standard template library. (3 hours)
- Fundamental computing algorithms – simple searching and sorting (linear and binary search, selection and insertion sort). (6 hours)
- Projects, exams and review. (6 hours)

Text Book and Other Required Materials: “C++ How to Program” by H.M. Deitel and P.J. Deitel.

Course Objectives and Their Relationship to Program Objectives:
A student should understand (i) object-oriented programming concepts and techniques, (ii) the principles of software engineering in object-oriented languages, and (iii) the fundamentals of programming in C++. A student should be able to design and implement object-oriented software to solve moderately complex problems. A student should master modern tools for computer-aided software engineering (CASE) and be able to write good program documentation. [Program Objectives this course addresses: 1, 2, and 4.]
Course Outcomes and Their Relationship to Program Outcomes:
The following are the course outcomes and the subset of Program Outcomes (numbered 1-11 in square braces "[ ]") they address:

16. Design and implement structured, robust, maintainable object-oriented programs from the specifications developed. [2,3,5,11]
17. Develop teamwork and management skills to divide tasks and effectively develop software in teams of 3 or more people. [4,7,9]
18. Produce well-documented code and program documentation for others. [3,5,7].
19. Use CASE tools for object-oriented software engineering such as syntax-aware and programmable editors, debuggers, and makefiles. [2,3,5,11]

Contribution of Course to Meeting the Professional Component
Engineering Topics: 100%

Computer Usage:
Students use PC and/or Unix workstations with GNU C/C++ compilers and debuggers. All of the assignments use computers as this is a programming course. The course also makes use of Internet services such as email and the web, for references. The course has a web site, which has downloadable software and documents, as well as reference links.

Design Credits and Features:
EE 205 has 1 design credit. All of the homework assignments and projects require program writing. Programming assignments are done individually while programming projects are done in groups of 2 or 3 students, with attention given to diversity within each group.

Instructor(s): T. Dobry, D. Yun

Person(s) Preparing Syllabus and Date: T. Dobry, December 10, 2008