COURSE DESCRIPTION

Department and Course Number: COMP 3000
Course Title: Object-Oriented Programming for Engineers and Scientists
Total Credits: 3
Required: No
Elective: No
Prerequisites: Departmental approval
Class meetings per week: 3 hours
Lab meetings per week: 0 hours
Course Coordinator: Dr. Homer Carlisle
Date Prepared: February 19, 2004

Current Catalog Description:
Fundamentals of object-oriented design and programming principles; data abstraction, identifying objects, problem decomposition, design and implementation of classes. Credit for the course will not be given to CSCI and SWEN majors.

Textbooks:

References:
None.

Course Objectives:
1. Be able to apply an iterative, incremental software development process.
2. Be able to apply good software engineering principles.
3. Be able to apply concepts and techniques of object-oriented design and programming.
4. Be able to perform fundamental testing and debugging activities.
5. Demonstrate an understanding of objectives 1 through 4 by developing full project folders (containing requirements analysis, design, source code and testing sections) for C++ programming projects throughout the semester.
6. Understand and be able to explain how techniques from this class can be applied to the development of significantly larger computer programs.
7. Be able to discuss fundamental time and space implications of various design decisions and C++ programming constructs.

Prerequisites by Topic:
None.

Topics Covered: (specify number of hours on each)
1. Fundamental computer concepts: hardware, integrated development environment (3 hours)
2. Fundamental programming language concepts (1.5 hours)
3. Software engineering principles, spiral software development and "Extreme Programming" (3 hours)
4. Critical differences between the "real world" and the computer domain (1.5 hours)
5. Fundamental C++ constructs: "Hello, World" through control constructs (3 hours)
6. Functions (4.5 hours)
7. Arrays (1.5 hours)
8. Object-oriented concepts and benefits (1.5 hours)
9. Fundamental C++ classes and objects (6.0 hours)
10. Dynamic memory allocation and pointers (4.5 hours)
11. Inheritance of operations (1.5 hours)
12. Polymorphism (3 hours)
13. The Standard Template Library (3 hours)
14. Recursion (1.5 hours)
15. Exception handling (3.0 hours)
16. Exams (3 hours)

Laboratory Projects: (specify number of weeks on each)
1. Main program shell (1 week)
2. MPG computation: Input, output, mathematical computations, control constructs (1.5 weeks)
3. Statistical computations: Library and user-defined functions (1.5 weeks)
4. Communications of the ACM journal article review and summary (1 week)
5. Classes: Inheritance and polymorphism (2 weeks)
6. Classes, pointers, the STL and polymorphism (2 weeks)

Oral and Written Communications:
All students are required to develop and demonstrate program documentation skills in the Project Folder required for each programming project. All students also demonstrate written communication in their review and summary of the article from the Communications of the Association of Computing Machinery (ACM).

Social and Ethical Issues:
Social and ethical issues are discussed during the semester when we discuss "real world" examples to emphasize various course topics (e.g., the difference in what constitutes "sufficient" testing for a program based on the type of software being developed).

Theoretical Content:
None.

Problem Analysis and Solution Design:
All students apply good software engineering principles and an iterative, incremental development process to all programming assignments. They must analyze requirements, design, implement (in C++), and test their programs, and document their results (including, when appropriate, intermediate results from different iterations) in their project folders.