Modularity and Data Abstraction in C++

CS {410,510} MD1, Winter 95

(Revision Dec 12, 1994)

Instructor:	Sergio Antoy
Office:	PCAT 115, ph. 725-3009, antoy@cs.pdx.edu
Office Hours:	Mon 9-10 (PSU), Tue 5:30-6:30 (OCATE), Thu 1:15-2:15 (PSU)
Time:	Tue 3-5:30 PM
Location:	19500 NW Gibbs Dr., Beaverton, and Newberger Hall 59, Campus

Subject

A comprehensive introduction to programming in C++ programming with emphasis on abstraction, encapsulation, modularization, and object orientation. Some advanced topics and techniques.

Textbook

C++ Primer (2nd ed.), S.B. Lippman, Addison-Wesley 1991, 0-201-54848-8.

Optional Reference

Margaret A. Ellis, and Bjarne Stroustrup, "The Annotated C++ Reference Manual (ANSI Base Document)", Addison-Wesley, 1990, ISBN 0-201-51459-1.

Prerequisites

Solid knowledge of algorithm, data structures, and fundamentals of programming languages.

Goals

Typical C and Pascal programmers must learn new concepts and ways of thinking for effectively using the language. The course provides these concepts and the opportunity to use them in a set of increasingly complex problems.

Grading

The basis for grading is the ability to use C++ in an appropriate way to solve problems for which C++ is an appropriate tool.

There are midterm and final exams, one sizable project in two phases, and some weekly-assigned programming tasks and problems useful for the project.

Homework, projects, and each exam have roughly equal importance. Work done later in the course has greater weight. NO late homeworks and project will be accepted without prior agreement with the instructor.

Resources

There are two C++ compilers on `rigel`, `CC` and `g++`, both with pros and cons. You are encouraged to write compiler independent code and use it with both compilers. Man pages and library information for both compilers are on-line.

Students are welcome to use their own compilers and systems, but some programming assignments and/or projects will be automatically processed by software tools which are on `rigel`. This requires compilable source code on `rigel`.

Programming

Factors accounted for in grading C++ code (in no particular order):

- Appropriate constructs (e.g., constants, classes, etc.)
- As simple as possible (but not simpler.)
- Efficient without sacrificing (too much) clarity.
- Appropriate algorithms and structures.
- Generality, without overdoing it.
- Correct output and output format.
- Comments and layout.
- Use of modularization and abstraction.
- Use of inheritance.

Miscellaneous

University rules and standard procedures apply to items not specifically discussed in this syllabus, such as cheating and protection, makeup exams, incomplete grades, etc.

A challenging project will be assigned on the first day of class. Students are strongly encouraged to have both the **textbook** and an active **computer account** by the first day of class. Check with the PSU (725-4036) or the OCATE (690-1460) office for the procedure to get a computer account.