Dr. Dichter
CS 400

Spring 2003
University of Bridgeport

Office: Technology Building 223
Office Hours:
Phone: 576 - 4763

email: dichter@bridgeport.edu
Web page: www.bridgeport.edu/~dichter

Required Text:

- Paul Wang, *C++ With Object Oriented Programming, Brooks/Cole*, 2001
  ISBN 0-534-37131-0

Reference Texts

- Johnsonbaugh and Kalin, *Applications programming in C++*, Prentice Hall, 1999
- Julius Dichter, *UML Introduction, OO Analysis and Design (class notes on website)*

This course is an advanced study of the C++ programming language. The emphasis will be on developing software which is dependable and reusable. Therefore a modular and library-oriented approach will be used. We will study the principles of Object Oriented programming. An important aspect will be the analysis and design of OO systems, OOA+D. We will study and apply classic and modern approaches to OO program construction. We will use class design techniques, including CRC card design and Unified Modeling Language (UML) approaches. The course will cover all major aspects of structured programming including data types, top-down design and object-oriented techniques, class design, control structures, recursion, scoping rules, random access files, higher-order functions, templates, namespaces, dynamic type detection, and exception handling. We will cover static and dynamic memory allocation. Dynamic allocation will be complemented with typical applications, such as lists. Common ADTs such as stacks, queues, lists and trees will be examined as they are implemented in the Standard Template Library, STL. We will examine the nature and applications of recursion including some classic sorting algorithms. We will also study the application of programming in C++ to web-based applications. Basics of CGI scripts, HTML forms will be covered.

The language will be C++. There will be several programming and design assignments to implement the above-described concepts. All assignments must be submitted in order to pass the course. Late assignments will be penalized *two* letter grades per week. In addition to the machine assignments, there will be **midterm** and **final** exams. Each of the two exams will count as 45 percent of the final grade. The programming assignments will constitute the **remaining portion** (10 percent) of the final grade.

CS400 - Topics

Classes and Objects. Class members and constructors. Inline methods, I/O streams. Details of C++ syntax.

OO Analysis and Design methods: Class identification, Class Design, System Design and Testing, The UML language.

Overloading functions, function signatures, declarations and definitions, references, dynamic storage management.

2-D and Higher-Dimension Arrays. Arrays of Pointers and Pointers to Arrays. Double Indirection, Dynamic allocation of high-order arrays, Command-Line Parameters. Advanced *typedef* applications. Defining Macros, Conditional Compilation

Pointers and address arithmetic, multiple indirection, generic programs, void * type, internal/external decoupling, the host object, the host pointer,

Destructors, *friends* of a class, recursive structures, static members.

Operator overloading, defining stream I/O for objects, iterators, pointers to members.

Midterm Examination

String operations, I/O stream library, inheritance and class derivation. Applications of bit level operations.

Multiple inheritance, virtual functions, Polymorphism.

OO Analysis and Design methods: Class identification, Class Design, System Design and Testing, The UML language.

Web programming: CGI's, HTML FORM element, UNIX environment

Functions with an unspecified number of parameters. Designing generic modules. Simple template functions and classes in C++. Using Files: Character, String, and Block I/O. Formatted I/O. I/O without conversions. Random and Sequential File Access. C++ I/O hierarchy.

Function templates, class templates, derived class templates, STL

Final Examination