

Julius Dichter  
CS 400

Spring 2007  
University of Bridgeport

Office: Tech 223  
Office Hours: M 2:30-3:30, T 10:00-12:00  
Phone: 576-4763

[www.bridgeport.edu/~dichter](http://www.bridgeport.edu/~dichter)  
[dichter@bridgeport.edu](mailto:dichter@bridgeport.edu)

**Overview:** 3 credit hours

This course introduces the modern object-oriented programming philosophy using C++ to the beginning graduate students. The emphasis is on developing the programming thought process in terms of objects and their interactions to each other. Concepts covered include data hiding, code reuse through inheritance, polymorphism, templates and exception handling, developing appropriate class hierarchy and code maintenance for large software projects.

**Prerequisites:** CS 102 or equivalent background.

**Course Objectives:** Having successfully completed this course, the student should be able to:

- Comprehend and apply proper object oriented programming concepts
- Use UML to model static and dynamic views of object oriented design
- Design and develop software using object oriented programming techniques
- Employ both C++ fundamental and intermediate C++ concepts
- Incorporate medium complexity data structures and algorithms in designing software
- Develop programs using standard libraries like STL and MFC

**Required Text:** Richard Johnsonbaugh and Martin Kalin, *Object-Oriented Programming in C++*, Second Edition, Prentice Hall, 2000.  
[ISBN-0-13-015885-2]

**Required Text:** Paul Wang, *C++ With Object Oriented Programming*, Brooks/Cole, 2001. [ISBN 0-534-37131-0]

**Class Topics:**

- Object Oriented Programming and Object-Oriented Design: Evolution, Features, Goals, Advantages (2.0 weeks)
- Principles of Modeling and Unified Modeling Language: Elements and Relationships (1.0 week)
- C++ Fundamentals: Data Types, Parameter Passing, Function Overloading, Inline Functions, Arrays and Pointers, Dynamic Memory Allocation, Memory Leaks, Standard, String and File I/O, Namespaces – Chapter 2 (1.5 weeks)
- Error Handling/Recovery: Assertions, Exception Handling – Chapter 2 (0.5 weeks)
- Classes and Objects: Constructors, Destructors, Member Methods, Overloading Operators – Chapter 3, 6 (1.5 weeks)
- Inheritance and Class Derivation: Derivation Principles, Multiple Inheritance – Chapter 4 (2.0 weeks)
- Polymorphism: virtual functions, pure virtual functions, run time type identification – Chapter 5 (2.0 weeks)
- Introduction to Microsoft Foundation Class (MFC) Library: introduction to event driven programming, message maps, creating GUIs, using AppWizard and ClassWizard – Chapter 9 ( 2.5 weeks)
- Templates: template functions and classes; and, Standard Template Library: containers, algorithms and iterators – Chapter 7 (1.0 week)

Other relevant sources shall be used to supplement the book material for various topics.

**Assignments:**

Programming Assignments – There will be several programming assignments throughout the semester.

*All assignments must be submitted in order to pass the course. Cheating will not be tolerated, and any student who cheats on any class assignment will fail the class. All programs are due at the beginning of the class. Late work will lose one letter grade per day.*

**Evaluation:**

Midterm (30%), Final Exam (40%), Programming Assignments and Projects (30%).

**Final Exam:**

TBA