CSCI 121 – Spring 2007
Computer Science II

Ben Coleman
coleman@cs.moravian.edu
214 PPHAC
Office Phone: 610-625-7781

Office Hours: MW 9:00 – 10:00
R 9:00 – 11:00
or by appointment
AIM: bjcoleman15

Course Description

A continuation of Computer Science I with emphasis on data and procedural abstraction. A study of the representation of data as realized in both hardware design and software development. Topics include advanced C++ language features, object-oriented programming, and encoding schemes for data.

Goals

- Write programs in C++ using objects, templates, operator overloading, and other advanced language features.
- Design programs using object-oriented design, the Standard Template Library, and recursion.
- Analyze the performance of algorithms using asymptotic notation, recurrences, and mathematical sums.
- Represent data in binary, two's complement, floating point notation, and other representations.

Required Text

In addition to the following required text, supplementary readings will be given periodically during the semester.

- *Problem Solving with C++* sixth edition, by Walter Savitch

You should expect to spend at least an hour before each class session working through the readings. This means reading the text for detail, studying the syntax for new language features, and working to learn vocabulary – not just skimming through the material before class.

Responsibilities

Your attendance is expected at each class meeting. You are also responsible for the contents of reading assignments, handouts, class activities, and class email.

If you have a disability that may affect your participation in this course, please contact me immediately to discuss academic accommodations.

Graded Materials

- Homework will be assigned nearly every class session. Some problems will be traditional pencil and paper work and others will be small programming assignments. These assignments will be due the next class session, and you should bring your solution to class. Because homework will be graded and returned the next class, you may not submit late work. You should expect to spend a half hour to a full hour on each homework assignment.
- Each Thursday class session will be used for a lab session. Labs will introduce a variety of tools and explore the current material from the classroom sessions. Lab work will be due each Monday at 11:59 p.m. Each lab will include a post-lab. In addition to the time spent in lab, you should expect to spend one to two hours completing each lab.
- Five projects will be assigned during the semester. These assignments emphasize all aspects of program development, including design, implementation, testing, and documentation. You will have approximately two weeks to work on each project, but it is especially important that you begin these assignments as soon as they are announced. Late submissions will be penalized ten percent per day. This deduction *may* be avoided by discussing your progress with me *before* the deadline.

	Date Distributed	Due Date
Project #1	Friday, January 26	Monday, February 12
Project #2	Wednesday, February 14	Friday, March 2
Project #3	Monday, March 12	Monday, March 26
Project #4	Wednesday March 28	Wednesday, April 11
Project #5	Friday, April 13	Wednesday April 25

- Three tests will be given during the semester on Friday, February 9, Friday, March 2, and Friday, April 13. Each test will be given in class, and may only be re-scheduled for medical or family emergencies. In such a situation, you must see me *before* the time of the test.
- The final will be cumulative and will be given in class during its schedule time.

Academic Honesty

Except on tests, you are encouraged to discuss the material and work with other students in the course. Specifically, on homework, labs, and projects, you may discuss any portion of the assignment with your fellow students. However, you must produce your own write-up of the material, and you are ultimately responsible for the material on the tests.

Grading

Homework	15%
Projects	25%
Labs	20%
Tests	25%
Final	15%

All grades will be calculated on the standard scale using pluses and minuses.

CSCI 121: Computer Science II

Spring 2007 Schedule

Date	Reading(s)	Topic(s)
M Jan 15		• First Day Activities
W Jan 17	• pp. 300-313; 326-435	• Using Files (Review)
R Jan 18		• Emacs / Shell Commands • File Processing • Introduction to debugging with `gdb`
F Jan 19	• pp. 410-416 • pp. 449-456 • pp. 467-484	• Partially-Filled Arrays (Review) • C-strings (Review) • Type `string` (Review)
M Jan 22	• pp. 540-555; 562-577 • pp. 682-697	• Classes (Review) • Separate Compilation
W Jan 24	• pp. 596-611; 614-619	• Friend Functions • The `const` modifier
R Jan 25		• Objects
F Jan 26	• pp. 619-637 • pp. 993-994	• Operator Overloading • **Project #1 Out**
M Jan 29	• pp. 500-512 • pp. 513-519	• Pointers • Dynamic Arrays
W Jan 31		• More on Dynamic Arrays
R Feb 1		• Dynamic arrays • More Debugging
F Feb 2	• pp. 645-657	• Copy Constructor, Destructor
M Feb 5	• pp. 658-661	• Assignment Operator
W Feb 7	• pp. 718-723	• Linked Lists - Introduction
R Feb 8		• Classes with Dynamic Memory
F Feb 9		• Test #1
M Feb 12	• pp. 724-737	• Linked Lists - Insertion • **Project #1 Due**
W Feb 14	• pp. 740-743	• Linked Lists - Deletion • **Project #2 Out**
R Feb 15		• Linked Lists
F Feb 16	• Malik pp. 548-557	• Algorithm Analysis • "Big-Oh" Notation
M Feb 19		• More "Big-Oh"

Date	Reading(s)	Topic(s)
W Feb 21	• TBA	• Sorting Overview
R Feb 22		• Sorting Part I
F Feb 23	• Malik pp. 568-575	• Sorting Analysis
M Feb 26	• TBA	• Sorting
W Feb 28	• TBA	• Sorting
R Mar 1		• Sorting Part II
F Mar 2		• Test #2 • **Project #2 Due**
M Mar 5 – F Mar 9		• Spring Break
M Mar 12	• pp. 893-904	• Function Templates • **Project #3 Out**
W Mar 14	• pp. 905-914	• Class Templates
R Mar 15		• Templates
F Mar 16	• pp. 484-493	• Vectors
M Mar 19	• pp. 923-937	• Iterators
W Mar 21	• pp. 938-952	• Containers
R Mar 22		• STL
F Mar 23	• pp. 953-968	• Generic Algorithms
M Mar 26	• TBA	• OpenGL • **Project #3 Due**
W Mar 28	• pp. 806-829	• Inheritance Basics • **Project #4 Out**
R Mar 29		• More STL
F Mar 30	• pp. 829-833	• Inheritance Details
M Apr 2	• pp 833-848	• Polymorphism
W Apr 4		• OO Design and Programming
R Apr 5		• Object-Oriented Programming
F Apr 6 – M Apr 9		• Easter Break
W Apr 11	• pp. 765-778	• Recursive Functions • **Project #4 Due**
R Apr 12		• More OO Programming
F Apr 13		• Test #3

Date	Reading(s)	Topic(s)
M Apr 16	• pp. 778-783	• Recursion • **Project #5 Out**
W Apr 18	• pp. 783-797	• Thinking Recursively
F Apr 20	• TBA	• Introduction to Data Representation
R Apr 19		• Recursion
M Apr 23	• TBA	• Integer Representations
W Apr 25	• TBA	• Floating Point Representations • **Project #5 Due**
R Apr 26		• TBA
F Apr 27		• Review