Syllabus:  CS503 Advanced Programming I  
CS305 Computer Algorithms I

Course Number: CS503-50/CS305-50
Course Title: Advanced Programming I/Computer Algorithms I
Instructor: Richard Scherl
Office: Howard Hall C-6
Phone: 571-4457
Email: rscherl@monmouth.edu
Web: www.monmouth.edu/~rscherl/

Class Times: Tuesdays 6:00 –7:50  Thursdays 6:00 - 7:50
Class Location: Edison 113
Office Hours: Tuesdays 4:00-5:30, Thursdays 4:00-5:30

Texts:

*Required* C++ An Introduction to Data Structures by Larry Nyhoff. Prentice-Hall, 2005 (Second Edition)

*Required* Some C++ text. For example, C++ How to Program by Deitel and Deitel, Prentice Hall. (4th edition or later)

Expected Work: Regular reading assignments, two midterm examinations, quizzes, a number of homework (primarily programming) assignments, and a final examination.

Organization Both CS305 and CS503 will meet together, but will be graded separately. The assignments may differ. The exams will be different.

Class Information All computer-generated overheads and handouts will be put on Ecampus.

Grading:

Midterms 30 %  
Quizzes/Class Participation 10 %  
Final 30 %
Homeworks 30%

Exam Dates

Midterm I  To be announced
Midterm II  To be announced
Final  During Finals Week

Class Participation: If you miss a class, it is your responsibility to find out about any announcements made in class, and about the material covered. Similarly you are responsible for all information included in any assignments whether handed out or transmitted online and for all the information in this syllabus. Class participation is strongly encouraged. Feel free to ask questions. When in doubt, ASK.

Sessions of the class held on Ecampus are required. You will be graded for your participation in the Ecampus activities.

Late Policies: Homeworks should be handed in on the date due. The deduction for late homeworks is 5% per day up till 1 week late. Late homeworks may not be handed in by email except by special arrangement under special circumstances. They should be handed in to me directly. If you leave them in my mailbox or under my door, you should also send me an email saying that you left it. They may also be mailed in by U.S. mail with the postmark date being used as the date handed in. After the one week has ended, late homeworks can be corrected, but will not receive credit as the solutions will have been discussed in class.

Computer: All students will need an account on rockhopper. You may do your assignments on any machine (including your own PC), but I can only guarantee that the software will work as intended on rockhopper. Additionally, all code must work with the g++ compiler on cslab.

Ecampus: The course will make use of Ecampus (http://ecampus.monmouth.edu) which is Monmouth’s online education software. All registered students will be automatically given accounts. You use your hawkdom password to access ecampus.

Prerequisites: CS176 with a grade of C or better or CS501B with a grade of B- or better. Familiarity with Unix.

Goals of the Course: Provide an in depth coverage of fundamental data structures and algorithms including linked lists, hashing, sorting, searching, stacks, queues, trees, and
graphs. Application to problem solving and object oriented design and implementation of moderate sized programs. Recursive definitions and algorithms are also covered. Additionally students are exposed to a number of more advanced topics from C++ including inheritance, templates, and an introduction to the Standard Template Library. The Eclipse development environment will be introduced and used during the course.

**Programming Assignments** There will be a number of programming assignments in C++. For every assignment, you must provide not only the code but also some sample runs of the code. When an assignment is completed, the following 2 things need to be done.

- Hand in a hard copy of both the code and the test runs in class.
- The code needs to be made available to me on rockhopper. You will need to create a directory to contain your final submissions. The directory and all of the files in it should be made readable by me but not by other students. Full instructions to be made available later.

The grade that a programming assignment receives will be based upon a variety of factors including

- Results of testing the code.
- Readability of the code.
- Ability of the author to answer questions about the code.

**Rules:** All cell phones need to be turned off during the class. If due to an emergency the phone needs to be left on, the permission of the instructor is needed.

During exams, all cell phones must be off and put away. They can not be kept on the desk. Additionally, no calculators may be on the desk during an examination. They are not needed.

**Academic Honesty:** Cheating in this course will not be tolerated. Both the giver and the receiver of information will receive the same penalty. The penalty is likely to be an F in the course and may very well lead to expulsion from Monmouth University. All such cases will be handled as outlined in the *Monmouth University Student Handbook*.

Homeworks may NOT be solved in collaboration. You may talk about problems with each other. Where does talking end and cheating start? My rule of thumb is: you may not have a pen/pencil in your hand while you are talking (and no keyboard!).

**Special Accommodations** Students needing accommodations are encouraged to see me during office hours or to make a specific appointment to discuss their needs. Students
with disabilities who need special accommodations for this class are encouraged to meet with me and/or the appropriate disability service provider on campus as soon as possible. In order to receive accommodations, students must be registered with the appropriate disability service provider on campus as set forth in the student handbook and must follow the University procedure for self-disclosure, which is stated in the University Guide to Services and Accommodations for Students with Disabilities. Students will not be afforded any special accommodations for academic work completed prior to the disclosure of the disability and prior to completion of the documentation process with the appropriate disability service office.

Regrades All disagreements about grading must be discussed in my office only. A request for an assignment or exam to be regrades must include a written note explaining the disagreement and also the original exam or assignment. These requests may be submitted in class or in my office. Regrade requests for a particular exam or assignment can only be accepted until the next test or assignment is due.

Course Content Tentative and subject to change

Segment 1 Review of Basic Concepts
- Program Development Process
- Abstract Data Types
- Classes and Objects
- Code Organization
- Functions
- Arrays
- Testing and Debugging

Segment 2 Intermediate Review
- Pointers and Addressing
- New and Delete Operators
- Using Dynamic Memory, “C-Style” Strings and Other Examples
- Copy Constructor
- Operator Overloading
- Memory Leaks and Destructor Calls
- Simple File I/O and Arguments for main()

Segment 3 Program Development
- Testing and Debugging
- Eclipse Development Environment
- Makefiles

**Segment 4** Some Basic Data Structures
- Stacks
- Queues
- Linked Lists

**Segment 5** The Standard Template Library
- Vectors
- Iterators
- Two Dimensional Vectors
- Queues, Lists, Deques and Ordered (Priority) Queues
- A List Application – Polynomials
- Lists vs. Vectors.

**Segment 6** Inheritance, Virtual Functions, and Dynamic Binding
- Derived Classes
- Constructors
- Object Oriented Design
- Virtual Functions and Dynamic Binding
- Application to Container Classes
- Potential Problems to Avoid
- Destructors – Testing for Memory Leaks

**Segment 7** Introduction to Trees
- Basic Tree Properties
- Binary Trees
- Tree Traversals
- A Binary Tree Class
- Building and Walking a Binary Tree
- Introduction to Recursion
- Simple Complexity Determination

**Segment 8** More Trees
- Recursive and non Recursive Traversals
- Binary Search Trees
- Heaps
- Balanced Trees
Segment 9 Graphs
- Some Definitions
- Some Properties of Graphs
- A Graph Class
- Representation for Graphs
- Graph Traversals
- Minimum Spanning Trees

Segment 10 Searching, Sorting, and Hashing
- Basic Complexity Issues
- Simple Search Algorithms
- Elementary Sorting Algorithms
- More Efficient Sorting Algorithms
- External Sorting
- Standard Library Algorithms
- Hashing

Initial Assignment  Chapters 1, 2, and 3 of Nyhoff.