CSS 342 A Data Structures, Algorithms, and Discrete Math I Spring 2014

http://courses.washington.edu/css342/zander/ **MW 5:45pm, UW1-102**

Professor: Dr. Carol Zander (zander@u.washington.edu)
Office: UW1-353
Phone: (425) 352-5276

Office Hours: M, W 3:15 – 4:15 pm, 8-9 pm
T 5 - 6 pm
or by appointment

Course description:
This sequenced course integrates mathematical principles with detailed instruction in computer programming. Topics include presentation of formal arguments to prove mathematical statements; development of algorithms; sorting and searching; algorithm analysis; object-oriented programming; basic abstract data types including lists, stacks, queues, and binary trees. Prerequisites: Calculus, two quarters programming

Grading:		
	Assignments	35%
	Midterm exam	30%
	Final exam	35%

A scale of 90s (3.5-4.0), 80s (2.7-3.4), 70s (1.8-2.4), 60s (0.7-1.7) is a guide although not strictly followed. Assignments consist of problems and programs.

Textbooks:
(1). *Data Abstraction & Problem Solving with C++*, Frank M. Carrano, Addison-Wesley
(2). *Discrete Mathematics and Its Applications*, Kenneth Rosen, McGraw-Hill
(3). A C++ book of your choice (optional)

Policies/Information:
Work is to be done independently unless directed otherwise; collaboration of work is NOT acceptable. You may discuss the problem statement with each other and help debug, but all designing and coding is to be done independently. Any pair programming must be approved by me. This class is run by honor code. By taking this class, you agree that you will not collaborate inappropriately on any work. In some cultures, family relationships and their loyalty are considered above all others. In this course, we are an academic family and you betray the instructor's and university's trust should you violate the honor code. This violation will be taken seriously. For the Student Conduct Code, see: http://www.uwb.edu/academic/policies/Academic_Conduct.xhtml

Computer use during lecture is limited to taking notes. No social networks, email, games, etc. You are not allowed to display any images on your screen during lecture as it is distracting. No cell phones. And keep whispering to a minimum so as not to distract others.

Pay attention to the catalyst due date. And recall catalyst can be slow, so don't wait for the last minute. Assignments will only be accepted via catalyst. Unless we have spoken about the circumstances and prior arrangements have been made, an assignment not turned in receives a grade of zero. Assignments will have a due time. If you turn in an assignment a few minutes after the due time, it will not be considered late. Catalyst will be open for an extra hour. Assignments turned in during that time lose 10%. If you email asking me to please not mark it late, you lose 20%.

No make-up exams will be given except under exceptional circumstances.

Because of limited time, all assignment questions will be answered on the message board. Implementation questions about your program should come to me in email. Also, there will be no debugging of code during class break. You can ask questions, but without your computer.

To request academic accommodations due to a disability, please contact Disability Resources (DRS) at 425.352.5307, 425.352.5303 TDD, or drs@uwb.edu. Provide documentation by the third week of the quarter.

Course goals:
The overall goal of CSS 342 is to learn discrete mathematics concepts and computer programming. You solve mathematical problems, present formal mathematical arguments, and program solutions to problems. You review searching and sorting algorithms, object-oriented programming, basic abstract data types, and study algorithm analysis. The programming language C++ is studied. Good software engineering techniques are used throughout. As with most technical courses, besides ability and motivation, it takes time and a lot of hard work to learn and master the subject. Expect to spend an additional 15 hours a week outside of class time on average.

Assignments:

– Follow all directions for turn-in on an assignment (found on course website, assignment's page). Turn in via catalyst dropbox linked from the course website. Code must compile and run correctly using the unix/linux g++ compiler.

– Syntax errors and run-time errors with not much output yield a low grade. Run-time errors or incorrect answers will result in a significant number of points being deducted from your grade.

If your code does not compile, run, and give CORRECT output for the sample main given, the highest grade you will receive on the assignment is D+ and the program will receive little feedback. If you do not put in the time to write it, time will not be put in to give you feedback on it.

Otherwise, you will be graded on documentation (clarity and completeness), style (indentation and use of blank lines/spaces), meaningful identifier names, organization of your program (modularity/design), efficiency (no useless, unnecessary, or unnecessarily complicated code), output (clarity and format), the overall readability, and following directions. Coding / documentation style guidelines and a detailed grading rubric can be found linked off the course website.

Topics covered and tentative 342 schedule:

This is an approximate ordering of topics. Some content will take more time, some less. Also, not all sections in all chapters will be covered; use topics as a guide. (Readings are labeled C++ or Math, corresponding to your two texts.) Although the C++ Interludes contain valuable content, they are not listed.

Week	Date	Topic	Reading	Assignments
1	Mar 31	Introduction, Preliminaries, etc. C++ language structure, structs, arrays	Website "C++ Notes", C++ appendix A	
	Apr 2	Searching, simple sorting (selection, bubble, insertion)	C++ 9.2 (5^{th}) (some of), 11.1 (6^{th})	
2	7	Data Abstraction (Objects and Classes), C++ classes	C++ 3 (5^{th}), 1 (6^{th})	
	9	More C++ classes	Website "C++ Notes"	Hw/lab 1 due
3	14	Pointers, Linked Lists	C++ 4.1-4.4 (5^{th}), 8-9 (6^{th})	
	16	Pointers, Linked Lists continued		Hw/lab 2 due
4	21	Templates, STL intro	C++ 4.5, appendix E (5^{th}), Interlude 7 (6^{th})	
	23	Algorithm analysis	C++ 9.1(5^{th}), 10 (6^{th}) Math 2.4, 3.1-3.3	
5	28	Algorithm analysis continued		
	30	Recursion	C++ 2, Math 5.3-5.4 (7^{th}), 4.3-4.4 (6^{th})	Hw/lab 3 due
6	May 5	**Midterm exam**		
	7	Sorting (using recursion)	C++ 9.2 (some of, 5^{th}), 11.2 (6^{th})	
7	12	Binary Search Tree intro	C++ 10.3 (5^{th}), 15.3 (6^{th}) Math 11.1-11.3 (7^{th}), 10.1-10.3 (6^{th})	
	14	Object-oriented design, Stacks	C++ 1, 6	Hw/lab 4 due
	18	Last day to withdraw		
8	19	Queues	C++ 7 (5^{th}), 13.1-13.2 (6^{th})	
	21	Recurrence and Induction	C++ 5.3 (5^{th}), 5.4 (6^{th}), Math 5.1-5.2 (7^{th}), 4.1-4.2 (6^{th})	Hw/lab5 design due
9	26	Holiday – Memorial Day		
	28	Recurrence and Induction continued	C++ appendix D (5^{th}), Appendix E (6^{th}) Math 8.1, 8.3 (7^{th}), 7.1, 7.3 (6^{th})	
10	Jun 2	Representation of Integers, Mathematical Foundations	Math 4.2 (7^{th}), 3.6 (6^{th}) Math 1.1, 1.3-1.5 (7^{th}), 1.1-1.4 (6^{th})	
	4	Last day stuff		Hw/lab 5 impl. due
11	11	Final Exam – in class		