

CSS 342 A Mathematical Principles of Computing Spring 2013 MW 5:45pm, UW1-102

<http://courses.washington.edu/css342/zander/>

Professor: Dr. Carol Zander (zander@u.washington.edu) **Office Hours:** M, W 4 - 5 pm
Office: UW1-353 T 5 - 6 pm
Phone: (425) 352-5276 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%	A scale of 90s (3.5-4.0), 80s (2.7-3.4), 70s (1.8-2.4),
Midterm exam	30%	60s (0.7-1.7) is a guide although not strictly followed.
Final exam	35%	Assignments consist of problems and programs.

Textbooks:

- (1). *Data Abstraction & Problem Solving with C++* 5th ed, Frank M. Carrano, Addison-Wesley
- (2). *Discrete Mathematics and Its Applications*, 7th ed, Kenneth Rosen, McGraw-Hill
- (3). A C++ book of your choice (optional)

Some C++ Books:

- *C++ How to Program*, H.M. Deitel and P. J. Deitel, Addison-Wesley.
- *The C++ Programming Language*, Third edition, Bjarne Stroustrup, Addison-Wesley.
- *C++ In Plain English*, Brian Overland, Wiley.

References:

- *Effective C++ Second edition: 50 Specific Ways to Improve Your Programs and Designs; More Effective C++: 35 New Ways to Improve Your Programs & Designs*, Scott Meyers, Addison-Wesley.
- *Thinking in C++*, Bruce Eckel, Prentice Hall.

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 ONLY. No social networks, email, games, etc. You are not allowed to display any images on your screen during lecture as it is distracting. And, no cell phones.

Pay attention to the catalyst due date. And recall catalyst can be slow, so don't wait for the last minute. Late assignments receive a grade of zero (unless we have spoken about the circumstances and prior arrangements have been made). No make-up exams will be given except under exceptional circumstances.

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. You will need to provide documentation of your disability as part of the review process prior to receiving accommodations (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 any special directions for turn-in given on an assignment. Always turn in an electronic copy of your source code (**only .h and .cpp files**) and other files asked for using catalyst. The dropboxes for assignments are linked from the course web site.

– Your code should compile and run properly using the unix/linux g++ compiler.

– Coding and documentation style guidelines can be found linked off the assignments page at <http://courses.washington.edu/css343/zander/style.html>

– 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. In other words, CHECK YOUR ANSWERS!!!!!!

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. A detailed grading rubric can be found at <http://courses.washington.edu/css342/zander/gradingRubric.doc>

GoPost: You can post questions (and answers) on a discussion/message board linked off the course website.

Topics covered and tentative 342 schedule:

This is an approximate ordering of topics. Material will take about the allotted time and not all sections in all chapters are covered. Topics are labeled C++ or Math and correspond to the appropriate textbook.

<u>Date</u>	<u>Topic</u>	<u>Material</u>	<u>Lab Due</u>
Apr 1	Introduction, Preliminaries, etc. C++ language structure, C++ structs, arrays	Website C++ Notes, C++ appendix A	
	Software Engineering Principles, pair programming	C++ 1, Website Course Notes	
3	<u>Searching, simple sorting (selection, bubble, insertion)</u>	C++ 9.2 (some of)	
8	pair programming Data Abstraction (Objects and Classes), C++ classes	Website Course Notes C++ 3	
10	<u>More C++ classes</u>	Website C++ Notes	lab 1 due
15	<u>Pointers, Linked Lists</u>	C++ 4.1-4.4	
17	<u>Pointers, Linked Lists continued</u>	C++ 4.1-4.4	lab 2 due
22	<u>Templates, STL intro</u>	C++ 4.5, C++ appendix E	
24	<u>Algorithm analysis</u>	C++ 9.1 Math 2.4, 3.1-3.3	
29	<u>Algorithm analysis continued</u>		
May 1	<u>Recursion</u>	C++ 2, Math 5.3-5.4 (4.3-4.4 ed. 6)	lab 3 due
6	Midterm exam		
8	<u>Sorting (using recursion)</u>	C++ 9.2 (some of)	
13	<u>Binary Search Tree intro</u>	C++ 10.3, Math 11.1-11.3 (10.1-10.3 ed. 6)	
15	<u>Object-oriented design, Stacks</u>	C++ 1, C++ 6	lab 4 due
19	LAST DAY TO DROP A COURSE		
20	<u>Queues</u>	C++ 7	
22	<u>Recurrence and Induction</u>	C++ 2.1, 5.3, Math 5.1-5.2 (4.1-4.2 ed. 6)	lab 5 design due
27	<u>Holiday – Memorial day (school closed)</u>		
29	<u>Recurrence and Induction continued</u>	C++ appendix D, Math 8.1, 8.3 (7.1, 7.3 ed. 6)	
June 3	<u>Representation of Integers Mathematical Foundations</u>	Math 4.2 (3.6 ed. 6) Math 1.1, 1.3-1.5 (1.1-1.4 ed. 6)	
5	<u>Intro to Design Patterns, last day stuff</u>	Website Course Notes	lab 5 due
10	Final exam, Monday, in class		