CSS 343 Data Structures, Algorithms, and Discrete Math II — Fall 2017

`http://courses.washington.edu/css343/zander` M/W 3:30pm, UW1-030

Professor: Dr. Carol Zander
zander@u.washington.edu
Office: UW1-260 (open area)

Office Hours: M, W after class or 8 pm (If you tell me)
Tue5:30 pm
or by appointment

Course description:

This sequenced course integrates mathematical principles with detailed instruction in computer programming. Topics include development of algorithms; algorithm analysis; object-oriented programming; abstract data types including trees, priority queues, graphs, and tables; regular expressions and context-free grammars. Prerequisites: CSS 342 with a grade of 2.0 or better.

Learning Objectives:

- An understanding of trees, balanced trees, heaps, hash tables and their uses
- An understanding of the graph data structure and associated algorithms
- Ability to design and implement a complex object-oriented problem (using inheritance)
- An understanding of the formal notation for a programming language

Course goals:

Refining and extending the concepts and skills introduced in CSS 342, students develop competencies associated with problem-solving, design, testing, and programming techniques. Topics include ADTs, data structures, related algorithms, and object-oriented design & programming. Formal automata theory as it applies to programming languages is introduced. Good software engineering and algorithm analysis techniques are used throughout.

As with most technical courses, besides ability and motivation, it takes time to learn and master the subject. No one succeeds without practice! Expect to spend an average of 15 hours a week outside of class time for this course; some of you may spend more time, some less time.

Textbooks: (1). *Data Abstraction and Problem Solving With C++*, Frank M. Carrano, Addison-Wesley (6th ed.)
(2). *Discrete Mathematics and Its Applications*, Kenneth Rosen, Addison-Wesley (optional)

Grading:

Assignments	35%
Midterm exam	30%
Final exam	35%

A scale of 90s (3.5-4.0), 80s (2.7-3.4), 70s (1.8-2.6), 60s (0.7-1.7) is a guide, although not strictly followed. Assignments consist of problems, designs, and programs.

Policies/Information:

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. In other words, be considerate.

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. It is always better to turn in something than nothing. 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%. If you email me your code, you get a negative grade!

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

Because of limited time, most 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.

Academic Integrity:

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. Some assignments may allow pair programming. 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 the 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`
For Academic Integrity and Plagiarism Prevention Resources: `http://guides.lib.uw.edu/bothell/ai`

There is much public code out there, even for course assignments. You are not allowed to use or to even view others' assignment code and will receive a failing grade. A misconduct letter will be put in your record which can lead to expulsion. Nor are you allowed to post your assignment code to any public website, ever.

Disabilities and Veterans:

Access and Accommodations:

Your experience in this class is important to me. If you have already established accommodations with Disability Resources for Students (DRS), please communicate your approved accommodations to me so we can discuss your needs in this course. If you have not yet established services through DRS, but have a temporary health condition or permanent disability that requires accommodations (conditions include but not limited to; mental health, attention-related, learning, vision, hearing, physical or health impacts), you are welcome to contact DRS at 425-352-5307 or email uwbdrs@uw.edu . DRS offers resources and coordinates reasonable accommodations for students with disabilities and/or temporary health conditions. It is the policy and practice of the University of Washington to create inclusive and accessible learning environments consistent with federal and state law.

Veterans:

If you are a student who has served in our nation's military forces, if desired, please feel comfortable to confidentially self-identify yourself to me so I can help you make a successful transition from the military to higher education.

.

Assignments:

- Follow all directions for turn-in (found on course website, assignment's page). Turn in via catalyst dropbox (course website link). Code must compile and run correctly using the 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 about 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 343 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 (Rosen), corresponding to the two texts.) There is a free discrete math text linked off the website, but it doesn't have much in it of the 343 content. The website notes should be sufficient for the discrete math topics covered. Although the C++ Interludes contain valuable content, they are not listed. If anyone has an older edition, the 5th edition's content is left in the syllabus. (Note that the terms assignment, lab, homework all mean the same thing.)

Week	Date	Topic	Reading	Assignments
1	Sep 27	Preliminaries, Review, Tree introduction		
2	Oct 2	Huffman encoding, Binary Search Tree	C++ 10 (5th), C++ 15, 16 (6th) Math 11.1-11.3	
	4	Priority Queues, Binary Heaps	C++ 11.2 (5th), C++ 13.3, 17 (6th)	
	5			Hw/lab 1 due
3	9	Priority Queues, Binary Heaps continued		
	11	Graphs (Dijkstra Shortest Path)	C++ 13 (partial, 5th), C++ 20 (partial, 6th) Math 10.1-10.3, 10.6 (web: 10.1-2)	
4	16	Graphs continued (Depth/Breadth-first)		Hw/lab 2 due
	18	Graphs cont.		
5	23	Balanced Trees (AVL, 2-3, B-tree)	C++ 12.1, 14.3 (partial, 5th), C++ 19.1,19.2.1-2.3, 19.5, 21.3.3 (6th)	
	25	Balanced Trees cont., Retrieval (trie)		
	28			Hw/lab 3 due
6	30	Object-oriented Design/Programming	C++ 1.1-1.2 (5th, 6th), 1.3 (6th)	
	Nov 1	**Midterm exam**		
7	6	Inheritance/Polymorphism, Hash Table introduction	C++ 8.2 (5th), Interludes (6th) C++ 12.2-12.3 (5th), C++ 18.4 (6th)	
	8	Factory Design pattern	Sample code	
	10	Holiday – Veteran's Day		
8	13	Hash Tables		
	14	**Last day to drop**		
	15	Hash Tables continued		
9	20	Inheritance/Polymorphism under the hood, Design Review	C++ 8.2 (5th), Interludes (6th)	Hw/lab4 design due
	22	Languages, Regular expressions	Math 13.4 (Course notes)	
	23, 24	Holiday – Thanksgiving		
10	27	Finite Automata	Math 13.3 (Course notes)	
	29	Context-free Grammars	Math 13.1, C++ 5.2 (5th), 5.1 (6th)	
11	Dec 4	Turing Machines	Course notes	
	6	Last day stuff		Hw/lab 4 implementation due
12	11	**Final exam (in class)**		