# CSS 343 Data Structures, Algorithms, and Discrete Math II Spring 2015

`http://courses.washington.edu/css343/zander` M/W 3:30pm, UW2-131

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

**Office Hours:** M, W 5:30-6:30pm
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.

**Grading:**

| | | |
|---|---|---|
| Assignments | 35% | Assignments consist of problems and programs. |
| Midterm exam | 30% | A scale of 90s (3.5-4.0), 80s (2.7-3.4), 70s (1.8-2.6), |
| Final exam | 35% | 60s (0.7-1.7) is approximately used. |

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

**Policies:**

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 the university's trust should you violate the honor code. This violation will be taken seriously. (See http://www.uwb.edu/academic/policies/Academic_Conduct.xhtml )

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.

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.

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:**

Refining and extending the concepts and skills introduced in CSS342, 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. Expect to spend an average of 15 hours a week outside of class time for this course; some of you may spend more, some less time.

**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.

– Your code must compile and run properly 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 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.) Although the C++ Interludes contain valuable content, they are not listed.

| Week | Date | Topic | Reading | Assignments |
|---|---|---|---|---|
| 1 | Mar 30 | Preliminaries, Tree introduction | | |
| | Apr 1 | Huffman encoding, Binary Search Tree | C++ 10 (5th), C++ 15, 16 (6th) | |
| | | **TODAY ONLY (April 1) – meet in UW1-050** | Math 11.1-11.3, | |
| 2 | 6 | Binary Search Tree continued | | |
| | 8 | Priority Queues, Binary Heaps | C++ 11.2 (5th), C++ 13.3, 17 (6th) | Hw/lab 1 due |
| 3 | 13 | Priority Queues, Binary Heaps continued | | |
| | 15 | Graphs (Dijkstra Shortest Path) | C++ 13 (partial, 5th),<br>C++ 20 (partial, 6th),<br>Math 10.1-10.3, 10.6, (web: 10.1-2) | |
| | 17 | | | Hw/lab 2, part 1 due |
| 4 | 20 | Graphs cont. (Depth/Breadth-first) | | Hw/lab 2, part 2 due |
| | 22 | Balanced Trees (AVL, 2-3, B-tree) | C++ 12.1, 14.3 (partial, 5th),<br>C++ 19.1, 19.2.1-2.3, 19.5, 21.3.3 (6th) | |
| 5 | 27 | Balanced Trees cont., Retrieval (trie) | | |
| | 29 | Object-oriented Design/Programming | C++ 1.1-1.2 (5th, 6th), 1.3 (6th) | Hw/lab 3 due |
| 6 | May 4 | **Midterm exam** | | |
| | 6 | Inheritance/Polymorphism, | C++ 8.1 (5th), interludes (6th) | |
| | | Hash Table introduction | C++ 12.2-12.3 (5th), C++ 18.4 (6th) | |
| 7 | 11 | Hash Factory, Hash Tables | Sample code | |
| | 13 | Hash Tables cont, | | |
| | | Inheritance/Polymorphism under the hood | C++ 8.2 (5th), interludes (6th) | |
| | 17 | **Last day to drop** | | |
| 8 | 18 | Languages introduction, Design Review | | Hw/lab4 design due |
| | 20 | Regular expressions | Math 13.4, (Course notes) | |
| 9 | 25 | Memorial Day Holiday, no school | | |
| | 27 | Finite Automata | Math 13.3, (Course notes) | |
| 10 | Jun 1 | Context-free Grammars | Math 13.1, C++ 5.2 (5th), 5.1 (6th) | |
| | 3 | Turing Machines, Last day stuff | Course notes | Hw/lab 4 due - implementation |
| 11 | 8 | **Final exam (in class)** | | |