# CSS 342 D Data Structures, Algorithms, and Discrete Math I Fall 2016

**University of Washington Bothell, Computing & Software Systems, STEM**

`http://courses.washington.edu/css342/zander/` **TTh 3:30pm, UW1-202**

**Professor:** Dr. Carol Zander
zander@u.washington.edu
Office: UW1-260

**Office Hours:** T, Th 5:30 – 6:30 pm
T, Th 8 pm if you tell me to wait
or by appointment

**Overview:**
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

**Learning Objectives:**

- Design and implement object-oriented programs in C++ using good software engineering practices
- Write software using basic abstract data types, a variety of data structures
- Use recursion to solve problems using data structures
- Solve discrete math problems

**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 (optional)
(3). A C++ book of your choice (optional); Examples include:

- Walter J. Savitch, Problem Solving with C++, Addison-Wesley.
- Harvey M. Deitel and Paul J. Deitel, C++: How to Program, Prentice Hall.
- Stanley B. Lippman and Josee Lajoie., A C++ Primer, Addison-Wesley.

**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.

**Policies/Information:**
Computer use during lecture is limited to taking notes. No social networks, email, games, etc. You are not allowed to display <u>any</u> 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 note that catalyst can be slow, so don't wait for the last minute. Assignments will <u>only</u> 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, most questions about the assignment 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. 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
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 ofWashington 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 on it.

Otherwise, you will be graded on documentation (clarity and completeness), style (indentation and use of blank lines/spaces, NO tabs in code), 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 the texts. There is a free discrete math book linked off the website. Use the table of contents to find the topic, although not all topics are covered in the book. ) Although the C++ Interludes contain valuable content, they are not listed here.

| Week | Date | Topic | Reading | Assignments |
|---|---|---|---|---|
| 1 | Sep 29 | Introduction, Preliminaries, etc.<br>C++ language structure, structs, arrays | Website "C++ Notes", C++ appendix A | |
| 2 | Oct 4 | Searching, simple sorting<br>(selection, bubble, insertion) | C++ 9.2 (5th) (some of), 11.1 (6th) | |
| | 6 | Data Abstraction (Objects and Classes),<br>C++ classes | C++ 3 (5th), 1 (6th) | |
| 3 | 11 | More C++ classes | Website "C++ Notes" | Hw/lab 1 due |
| | 13 | Pointers, Linked Lists | C++ 4.1-4.4 (5th), 8-9 (6th) | |
| 4 | 18 | Pointers, Linked Lists continued | | |
| | 20 | Templates, STL intro | C++ 4.5, appendix E (5th), Interlude 7 (6th) | Hw/lab 2 due |
| 5 | 25 | Algorithm analysis | C++ 9.1(5th), 10 (6th) Math 2.4, 3.1-3.3 | |
| | 27 | Algorithm analysis continued | Course notes | |
| | 29 | | | Hw/lab 3 due |
| 6 | Nov 1 | Recursion | C++ 2 | |
| | 3 | **Midterm exam** | | |
| 7 | 8 | Sorting (using recursion) | C++ 9.2 (some of, 5th), 11.2 (6th) | |
| | 10 | Binary Search Tree intro | C++ 10.3 (5th), 15.3 (6th)<br>Math 11.1-11.3 (7th), 10.1-10.3 (6th) | |
| 8 | 15 | Trees continued | | Hw/lab 4 due |
| | 15 | Last day to withdraw | | |
| | 17 | Object-oriented design, Stacks | C++ 1, 6 | |
| 9 | 22 | Queues | C++ 7 (5th), 13.1-13.2 (6th) | |
| | 24 | Holiday – Thanksgiving | | |
| 10 | 29 | Recurrence and Induction | C++ 5.3 (5th), 5.4 (6th),<br>Math 5.1-5.2 (7th), 4.1-4.2 (6th) | Hw/lab5 design due |
| | Dec 1 | Recurrence and Induction continued | C++ appendix D (5th), Appendix E (6th)<br>Math 8.1, 8.3 (7th), 7.1, 7.3 (6th) | |
| 11 | 6 | Representation of Integers,<br>Mathematical Foundations | Math 4.2 (7th), 3.6 (6th), Course notes<br>Math 1.1, 1.3-1.5 (7th), 1.1-1.4 (6th), notes | |
| | 8 | Last day stuff | | |
| | 9 | | | Hw/lab 5 impl. due |
| 12 | 13 | Final Exam – in class | | |