

# CSS 501 Data Structures and Object-oriented Programming I

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

Fall 2015

<http://courses.washington.edu/css501/zander>

TTh 6-7:40 pm, UW1-051

**Professor:** Dr. Carol Zander

Office: UW1 – 337 (Geetha Thamarasu’s office)

Email: [zander@u.washington.edu](mailto:zander@u.washington.edu)

**Office Hours:** T, Th 4 – 5 pm (in UW1-337)

T, Th after class (UW1-051)

or by appointment

### Overview:

This sequenced course integrates mathematical principles with detailed instruction in computer programming, data structures and object-oriented programming. Students consider basic data types, their uses, and implementations. Students design solutions to programming problems using object-oriented techniques with various data types. Modern software engineering practices are used. Some discrete mathematics topics are covered.

### 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
- Evaluate the trade-offs between different data types and their implementations
- Solve discrete math problems

### Textbook:

Data Abstraction & Problem Solving with C++ , Frank M. Carrano, Addison-Wesley

You may want to have a C++ text as well. 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.

### 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 or help debug, but any actual work to be turned in, must be done without collaboration. 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 (for the student conduct code, see: [http://www.uwb.edu/academic/policies/Academic\\_Conduct.xhtml](http://www.uwb.edu/academic/policies/Academic_Conduct.xhtml) ).

There is much public code out there, even for course assignments (for example, dropbox). 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. Nor are you allowed to post your code to any public website ever,

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.

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](mailto:drs@uwb.edu). Provide documentation by the quarter’s third week .

### Assignments / Grading:

Homework /programming assignments:	35%
Midterm exam:	30%
Final exam:	35%

In general, a student who achieves 80% of the possible points should expect to receive a 3.0 course grade.

## Homework and exam policies:

Pay attention to the catalyst due date. Catalyst can be slow, so don't wait for the last minute. Assignments will only be accepted via catalyst. Do NOT email me turn-in code. Unless we have spoken about the circumstances and prior arrangements have been made, a missing assignment receives a grade of zero.

Written assignments or any hard copies are expected to be turned in before the start of lecture.

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

In general, 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.

## Tentative Schedule (subject to change, material coverage may run over)

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. The edition, 5<sup>th</sup> or 6<sup>th</sup>, is in parentheses. Although the C++ Interludes contain valuable content, they are not listed.

Week	Date	Topic	Reading	Assignments
1	Oct 1	Introduction, Preliminaries, etc. C++ language structure, structs, arrays	Website "C++ Notes", C++ appendix A	
2	6 8	Searching, simple sorting (selection, bubble, insertion) Data Abstraction (Objects and Classes), C++ classes	C++ 9.2 (5 <sup>th</sup> ) (some of), 11.1 (6 <sup>th</sup> ) C++ 3 (5 <sup>th</sup> ), 1 (6 <sup>th</sup> )	
3	13 15	More C++ classes Pointers, Linked Lists	Website "C++ Notes" C++ 4.1-4.4 (5 <sup>th</sup> ), 8-9 (6 <sup>th</sup> )	Hw/lab 1 due
4	20 22	Pointers, Linked Lists continued Templates, STL intro	C++ 4.5, appendix E (5 <sup>th</sup> ), Interlude 7 (6 <sup>th</sup> )	Hw/lab 2 due
5	27 29 31	Algorithm analysis Algorithm analysis continued	C++ 9.1(5 <sup>th</sup> ), 10 (6 <sup>th</sup> ), Course notes, Math text	Hw/lab 3 due
6	Nov 3 5	Recursion <b>Midterm exam</b>	C++ 2	
7	10 12	Sorting (using recursion) Binary Search Tree intro	C++ 9.2 (some of, 5 <sup>th</sup> ), 11.2 (6 <sup>th</sup> ) C++ 10.3 (5 <sup>th</sup> ), 15.3 (6 <sup>th</sup> )	Hw/lab 4 due
8	17 19	Trees continued <b>Last day to withdraw</b> Object-oriented design, Stacks	C++ 1, 6	
9	24 26	Queues <b>Holiday - Thanksgiving</b>	C++ 7 (5 <sup>th</sup> ), 13.1-13.2 (6 <sup>th</sup> )	Hw/lab5 design due
10	Dec 1 3	Recurrence and Induction Recurrence and Induction continued	C++ 5.3 (5 <sup>th</sup> ), 5.4 (6 <sup>th</sup> ), Math text C++ appendix D (5 <sup>th</sup> ), Appendix E (6 <sup>th</sup> )	
11	8 10	Representation of Integers Propositional, Predicate logic notation Last day stuff	Course notes Course notes, Math text	Hw/lab 5 impl. due
12	17?	<b>Final Exam – in class</b>		