

CS 311 - F01 (3 credits)
Data Structures and Algorithms
Spring 2014

Instructor: Dr. Chris Hartman
Email: cmhartman@alaska.edu
Office: 201-D Chapman
Office Phone: 474-5829 (email will *always* be better)
Office Hours: TWF 9:00 – 11:00
Prerequisites: CS 202 (Computer Programming II)
Schedule: MWF 11:45-12:45, Chapman 104
Midterm Exam: Friday, March 14th
Final Exam: 10:15 a.m. - 12:15 p.m., Friday, May 9

Text: There is *no* required text for this class. I will provide the slides, other reading materials, and web links. If you like to have a hard copy of a book to read, much of the material (especially in the latter part of the semester) is based on *Carrano, Data Abstraction and Problem Solving with C++: Walls and Mirrors* which is currently in the 5th edition. Older editions are just as good and much cheaper.

Course BlackBoard: <http://classes.uaf.edu>

Procedure: **Class meetings** will be the standard lecture/discussion format. Lecture slides will be available on Blackboard after each class. An archive of last year's slides is available. **Homework assignments** will be given roughly bi-weekly (8 total), and will consist of C++ programming. The first few homework assignments are to be done by each student individually. Later in the semester, most assignments will be done in groups of 2–3 students. At my discretion, late homework will be accepted for partial credit. **Exams** Two exams will be given: midterm and final. The final will be comprehensive, emphasizing material covered after the midterm. Questions will be drawn from the example questions (with answers!) available on Blackboard.

Grades: Assessment of the following items will be used in the following proportions to determine student grades.

Assignments	50%
Midterm Exam	25%
Final Exam	25%

Final course grade may be curved upwards based on total points.

Policies: Students are expected to be at every class meeting on time, and are responsible for all class content, whether present or not. If absence from class is necessary, in-class work and homework may be made up only if the instructor is notified as soon as possible; in particular, absences due to scheduled events must be arranged ahead of time.

Academic dishonesty will not be tolerated, and will be dealt with according to UAF procedures. In particular, homework turned in must be substantially your own work, although you are welcome to discuss methods of solution with other students.

Students in this class must pay the CS lab fee. Payment allows access to the Chapman 103 lab.

I will work with the UAF Office of Disability Services (208 WHITAKER BLDG, 474-5655) to provide reasonable accommodation to students with disabilities.

UAF academic policies: <http://www.uaf.edu/catalog/current/academics>

CS Department policies: <http://www.cs.uaf.edu/cs/Policies.html>

Catalog Description:

Data structures and the algorithms for their manipulation. Object-oriented programming, arrays, tables, stacks, queues, trees, linked lists, sorting, searching and hashing.
Prerequisites: CS F202. (3+0)

Course Goals:

Generally, we will learn how to write high quality robust C++ code, using several advanced features of the language. We will utilize this knowledge to investigate common data structures and well known algorithms.

Student Learning Outcomes:

After taking this class, you should:

- Have experience writing and documenting high-quality “Advanced C++” code.
- Understand how to write robust code with proper error handling.
- Be able to perform basic analyses of algorithmic efficiency, including use of “big-O” notation.
- Be familiar with various standard algorithms, including those for searching and sorting.
- Understand what data abstraction is, and how it relates to software design.
- Be familiar with standard data structures, including their implementations and relevant trade-offs.

The following topics will be covered, roughly in order:

- Advanced C++
- Software Engineering Concepts
- Recursion
- Searching
- Algorithmic Efficiency
- Sorting
- Data Abstraction
- Basic Abstract Data Types & Data Structures:
- Smart Arrays & Strings
- Linked Lists
- Stacks & Queues
- Trees (various types)
- Priority Queues
- Tables
- Other, as time permits: graph algorithms, external methods.

Advanced C++ consists of understanding:

- The structure of a package
- Parameter passing
- Operator overloading
- Silently written & called functions
- Pointers & dynamic allocation
- Managing resources in a class
- Templates
- Containers & iterators
- Error handling
- Exceptions
- Linked Lists