CS 111 - Data Structures
Syllabus and Policies

Professor Alistair Campbell
Taylor Science Center 2016, ext. 4377, acampbel@hamilton.edu
Office Hours: Mondays 9:00–10:30; Tuesdays 1:00–2:15, and by appointment

Course Description. This is the second course in the introductory sequence, intended for prospective computer science concentrators and other like-minded individuals. In this course you will learn a great deal about what happens behind the scenes in computational processes, particularly the low-level implementation of data structures and algorithms that work with them. We'll do implementation in the C++ programming language.

Prerequisite. Good work[1] in CS 110, or a score of 4+ on the AP Computer Science A exam, or instructor permission. You need to be very comfortable with variables, functions, basic classes and methods, if-statements, and while-loops. We expect that you can devise programs to implement moderately complicated algorithms, and that you can design algorithms for simple problems yourself.

Readings.

1. *Open Data Structures (in C++)*, by Pat Morin
   http://opendatastructures.org/ods-cpp.pdf
2. *C++ Language References and Tutorial*.
   http://cplusplus.com
3. Kevin Heard, *Unix Tutorial*.
   http://people.ischool.berkeley.edu/~kevin/unix-tutorial/toc.html
4. Piazza course page (resources, questions, and answers)
   piazza.com/hamilton/fall2015/cs111/home

Calendar. This will change as needed.

Week	*Dates*	*Topic*
1	8/27–9/5	Intro to C++
2	9/6–9/12	Abstract data types, stacks
3	9/13–9/19	Linked structures

[1] In my view, good work means being in the top half of your 110 class, probably having A- or better.

4	9/20–9/26	Linked stacks and queues
5	9/27–10/3	Recursion
6	10/4–10/10	Recursion
7	10/11–10/14	Recursion (Fall Break)
8	10/18–10/24	General Lists
9	10/25–10/31	Trees
10	11/1–11/7	Binary search trees
11	11/8–11/14	Priority Queues
12	11/15–11/21	Heaps
	Thanksgiving Break	
13	11/29–12/5	Hash tables
14	12/6–12/12	Sorting
15	12/13–12/19	Exam week

Evaluation. Your final course grade is **based on** the following. Failure in any one component is grounds for failing the course.

Homework & Projects	40%
Evening exams	30%
Final exam	20%
Academic Engagement	10%

You may appeal grading decisions no later than one week after we return the assignment to the class.

Your work will be as follows.

- Short pencil-and-paper homework will be assigned over weekends.
- Several programming assignments spanning 1, 2, or 3 weeks each.
- Three evening exams, which will be held Wednesdays: ***September 23***, ***October 21***, and ***November 11***, at 7:00 PM. If you have a scheduling conflict with another class, you will need to notify Prof. Campbell during the first week of the semester.
- A final exam, which will be held ***Friday, December 18, 9 AM–12 PM***. All students will take the final exam at this time. Do not make any plans that would prevent you from taking the final exam.

Attendance policy. You are expected to attend every class. You may be excused only for college-sanctioned activities. You must let Prof. Campbell know about such absences as soon as you are notified.

Late policy. Late work will not be accepted without Prof. Campbell's permission in advance.

Laboratory. The Department of Computer Science provides laboratory space, computer equipment, and software for your use in this course. You may only use the hardware and software that you have been authorized to use. We expect you to treat all equipment with the utmost respect and care. Modifying the configuration of any equipment without authorization is prohibited. Please report problems with labs or equipment to our department director of laboratories, Jerry Tylutki (Science Center 3020, `jtylutki@hamilton.edu`, x4289)

Honor code policies. All assignments are meant to be individual efforts. They are not pair- or group-projects. Therefore, you may not "work together" on your assignments.

However, you may *discuss* your assignments with each other, so long as it is a *verbal* discussion. There are three guiding rules.

1. While conducting your discussion, you may not consult written materials about the specific problem at hand beyond the problem statement. You may use general reference materials such as programming manuals.

2. You may use a whiteboard or paper during the discussion to help participants focus on the elements of the problem, but, after the discussion is over, the board must be erased or the paper destroyed. No copy of this work may be retained by anyone. Photographs of whiteboard work violate the policy.

3. The discussions are not noticeably frequent. It's not reasonable to put strict limits on talking about approaches, particularly in lab. But you should take care that your you should not be doing so in such a way that your work becomes practically identical. If you are collaborating about every detail, or "working in lock step" with each other, you have defeated the notion of individual effort.

After the discussion where any part of a solution is hashed out, you, the **individual student**, must work it out again, by yourself.

Academic work almost always builds upon the work of others. In order for me to properly evaluate your contribution to the finished product, you must use citations to acknowledge any outside source of help. All sources must be cited including, but not limited to, discussions with the professor, your peers, the course TAs, tutors, textbooks, class notes, and anything from the Internet. Questions about the proper form of citations should be directed to Prof. Campbell. In short, if you didn't write it all by yourself, you must provide a citation for each instance of collaboration.

It is not sufficient simply to write "worked with *person's name*" at the top of your work. In order to receive full credit, each citation must be tied to a particular part of the assignment and must (1) identify the source, and (2) describe the nature of the help received.

Here are two examples of proper citations.

```
CITE: Teaching Assistant Maggie Coleman
DESC: Suggested additional parameter to combine multiple functions into one

CITE: http://www.math.rutgers.edu/~greenfie/gs2004/euclid.html
DESC: Source of Euclid's method for determining greatest common divisor
```

Modern technology policy. Cell phones and similar electronic devices are not permitted in the classroom, laboratory, or professor's office. (They must be neither seen nor heard.) Personal laptop computers may not be used in class or lab without special permission (see Disability Policy below.) Audio or video recording of lectures is not allowed. Ear-buds/headphones are not appropriate. Accessing any material not related to this course is prohibited during class.

Disability policy. Any student with a documented disability requesting academic adjustments or accommodations must speak with Prof. Campbell during the first two weeks of class and provide written documentation of the suggested accommodation from the Dean of Students Office (Allen Harrison, Elihu Root House; ext. 4021). All discussions will remain confidential.

A Note about the Grading of Programs. Be advised that, public opinion notwithstanding, it is not the case that "a program is a program is a program." Just as many papers written for a particular assignment in an English class can and do vary in quality, so too do programs that address a particular problem. In fact, this may be one of the most important lessons that you will learn in this course. Questions like:

Is the program thoroughly documented to reflect your understanding of the problem? Were the algorithms implemented in efficient and concise manners? Are the algorithms generally correct? Did you choose data structures and algorithms that were appropriate to the problem being solved? Is the program "user-friendly?" are all legitimate in the context of grading a program. The more you attend to such questions in doing your work, the more your work will be rewarded. We will do our best to be consistent and fair, and to explain our evaluations thoroughly.

About your professor. I concentrated in computer science and mathematics at Colgate University (A.B.) and I studied artificial intelligence, knowledge representation, and cognitive psychology at the University at Buffalo (M.S., Ph.D.) My current research involves programming language implementation. This is my twenty-second year teaching computer science, the seventeenth at Hamilton College. My wife Colleen is a professional vocal musician and music teacher at Jarvis Middle School in Mohawk, New York, where she also serves as a teachers' union vice-president. We have an 11-year-old son named Aubrey, and a $7\frac{1}{2}$-year-old daughter named Evelyn. Some things I like to do when I'm not busy with my Hamilton job include Scouting (Assistant Scoutmaster with Troop 14 in Washington Mills), bicycling, brewing beer, and enjoying single-malt scotch whiskey.