

# Design Principles Syllabus

Professor Mark Bailey  
Science Center 3004, ext. 4229  
mbailey@hamilton.edu

Office hours: Monday 2:00–3:30, Tuesday 2:00–3:30, Wednesday 11:00–12:00, others only by appointment.

**Course Description.** We will start this course by covering object-oriented programming, and creation and use of classes. After that brief introduction, we will focus on a class of fundamental programming models and their theoretical foundations that make building large, reliable software feasible. These models, called data structures, are used for storing, organizing, and fetching data efficiently. We will study many of the traditional data structures, including linked lists, queues, stacks, trees, and hash tables. In addition, we will study the two most important classes of algorithms: searching and sorting. From time to time, we will digress into a variety of software engineering topics that will help you develop mature programming skills and problem solving abilities. Along the way, you will be expected to develop proficiency in another programming language: C++.

**Readings.** There are two required readings for this course:

- *Problem Solving with C++*, 10<sup>th</sup> Edition, by Walter Savitch  
ISBN (for rental): 9780134448282 (<https://tinyurl.com/cs102-book>)
- Course webpage: <https://tinyurl.com/cs102-fall125>

**Evaluation.** At the end of the semester, I will compute an average using the weights below. Failure of any of the parts is grounds for failing the course.

|                  |     |
|------------------|-----|
| Projects .....   | 25% |
| Exams .....      | 30% |
| Labs .....       | 10% |
| Codelets .....   | 10% |
| Final exam ..... | 25% |

I will use this average, as well as your level of intellectual engagement and my observations of your performance throughout the semester to inform my decision on your course grade. Therefore, your grade may be higher or lower than the average might suggest.

You may appeal grading decisions for individual assignments no later than one week after I return the assignment to the class.

**Calendar** (*highly tentative*).

| Week | Monday Date | Topics  | Due    |
|------|-------------|---|--------|
| 0    | Aug. 25     | Introduction to C++<br>[Chapters 1–6]           |        |
| 1    | Sep. 1      | C++ and Classes<br>[Chapter 10.2]               |        |
| 2    | Sep. 8      | Introduction to Recursion<br>[Chapter 14]       |        |
| 3    | Sep. 15     | Array-based Stacks<br>[Chapter 7, Chapter 13.2] |        |
| 4    | Sep. 22     | Pointers and Stacks<br>Chapter 9, 11.2          | Exam 1 |
| 5    | Sep. 29     | Dynamic Memory                                  |        |
| 6    | Oct. 6      | More Recursion                                  |        |
| 7    | Oct. 13     |   |        |
| 8    | Oct. 20     | Linked Lists<br>Chapter 13.1                    | Exam 2 |
| 9    | Oct. 27     | Linked Lists                                    |        |
| 10   | Nov. 3      | Binary Trees                                    |        |
| 11   | Nov. 10     | Sorting   | Exam 3 |
| 12   | Nov. 17     | Hash Tables                                     |        |
| 13   | Dec. 1      | Heaps   |        |
| 14   | Dec. 8      | Templates<br>[Chapter 17]                       |        |
| 15   | Dec. 15     | Final Exam                                      |        |

**Attendance.** I expect you to attend every class. I will excuse you only for college-sanctioned activities; you must let me know about such absences as soon as you are notified. I expect you to arrive to class on time. Chronic tardiness (more than one) will result in a significantly lower final grade.

**Collaboration.** For each collected assignment, you may work by yourself or with one other student. If you work as a pair, you must submit the assignment together. You may choose a different partner (or to go solo) on each assignment separately.

Beyond your partner on an assignment, you may discuss *ideas* with other students in the class, but you may not help others with their code or share code with them. **You are never allowed to copy any amount of code from another student or from other sources (except the textbook and your class notes), including the Internet.** Copying of code is a violation of the honor code.

**Use of automated assistants.** The use of any automated assistant (Github Copilot, Chat-GPT, etc.), whether it uses artificial intelligence or not, is not allowed for graded work in this class. Any violation of this policy will result in failure of the course. For ungraded work in this course, I strongly encourage you to resist its use since it eliminates any learning potential of the assignment.

**Extensions.** I'm quite supportive of extensions liberally. If you need an extension, just ask ahead of the due date. All I ask is that you share with me the reason you need the extension. It is possible to use this flexibility too much, but I will let you know ahead of time if you are approaching that point.

**Publishing your work.** You are not allowed to publish your work for this course during the semester or afterward. Publishing is using any technology or process that makes your work available to anyone else. This includes, among others, webpages, Google Drive, Facebook, GitHub, or any other technology that enables others to acquire your work. Making your work available to others is a violation of the honor code and will result in failure of the class and may be applied retroactively.

**Late policy.** Late assignments will not be accepted without my approval *prior* to the due date.

**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. Audio, video, or image recording of lectures, the white board, or projection screen is not allowed. Ear-buds/headphones are not appropriate.

**Department laboratories.** 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, David Deeley (Science Center 1005, ddeeley@hamilton.edu, x4452)

The Science building is the best place to do computer science work. Use the space outside of the Science 2013 suite, use the 2017 classroom anytime a class is not scheduled between 9:00 a.m and 4:00 p.m., bring your laptop and work in the Science atrium, or use the Introductory Computer Science Laboratory (Science Center 3040) when it is open.

**Citing other's work.** 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 me. 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 Josh Harmsen
// 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
```