

## CS-102 Design Principles: Project 8

---

**Due date:** Monday, May 1 at 11:00pm

This is the final project! You may not turn it in late unless you have an incomplete for the course. If you do not feel that your project is finished, you should still hand it in, and explain what your project is capable of in your video.

You are welcome to work jointly with one other person, and you may come up with your own goals. Alternatively, you can work on one of the ideas that will be presented later in the instructions.

There is no Github link to accept. To submit your project, you will share your code on Gradescope, along with a video explanation of your project and any other supporting files that you want to include. You may use any code that we have written together in class or posted on Discord, but not any code taken from anywhere else.

---

### Requirements and options

Whatever idea you go with, your design should use (for a good design reason) at least one class. Your project will be hard-pressed to get a grade above B unless it uses (for a good design reason) at least one abstract data type (e.g. stack, queue, hash table, priority queue, heap, binary search tree) OR unless it uses inheritance.

Your file structure should adhere to the standards set in the most recent project:

- ☐ Each class has its own `.h` and `.cpp` files; your class declaration should be in the `.h` file, and your method definition(s) should be in the `.cpp` file.
  - ☐ Put header guards in the `.h` file, using the convention `FILENAME_H`.
  - ☐ Compile all of the `.cpp` files to `.o` object files for each class.
  - ☐ Link in the object files when compiling your code.
- 

### Some ideas

Here are a few ideas for those of you who would prefer not to come up with a completely personal design. You will need to do a bit of research about them:

- ☐ Write a priority queue that is implemented with a min-heap, and use the priority queue to find a minimum spanning tree for a given graph.
- ☐ The New York Times is beta testing a new game called Digits; write a solver for that game.
- ☐ Implement a *skip list*, including methods for insert, lookup, and delete, using linked lists (from scratch).

- ☐ Implement the *nearest neighbor* heuristic algorithm for the *traveling salesperson problem*.
- ☐ Write a hashing algorithm for hash tables (e.g. the SHA-1 hashing algorithm).
- ☐ Use a breadth-first search to detect if a graph is *bipartite*.

## Grading

Your grade will come from:

60%	a working program meeting the guidelines
20%	style
10%	video presentation
10%	diary

**Style.** Your code must satisfy all of the requirements that we have introduced during this course: Use the conventions that we taught you in class for multi-file projects, use the conventions that we taught you in class for making classes and defining their methods, eliminate dead code, use correct programming constructs, correct indentation, appropriate spacing around parentheses, braces, brackets, and operators, follow our usual variable naming conventions, provide file and function headers that are correct and in the prescribed format. Each function in your code should fit in a window (so that they are about twenty lines long at most). Variables should be declared near their initial use. None of your lines of code should exceed 90 characters.

**Video presentation.** Your presentation should be carefully planned to fit in five to eight minutes. Part of that time should be devoted to showing us what your program does, and the rest should focus on how your project fulfilled the guidelines that we set. Do not show us your code during your video. You can record your presentation in a Zoom session, unless you have another method that you prefer. Alternatively, you can show the class what you did in lieu of creating this video, but you should let me know that you're interested in this before the due date.

**Diary.** You should post your progress (both in code form and written summary form) on Discord. When you post code, please drag your file(s) into Discord so that your progress up to that point is saved forever. If you think of another way to handle this, check with me first. Each time you post your code, you should briefly summarize what you have accomplished since the last update. To achieve a good diary grade, your diary must show that you are making consistent progress throughout the two weeks that you have to work with.