

CSCI 1300 Introduction to Computer Programming
Instructor: Fleming/ Gupta

Homework 8: Project Proposal

Due Monday, March 19th by 6 pm

Homework 8: Class files & Code Skeleton

Due Sunday, April 8th by 6 pm

Homework 8: Final Deliverables

Due Sunday, April 22nd by 6 pm (no bonus available)

Homework 8: Interview Grading

begins on Monday, April 23rd

Homework 8: Project Report

Due Sunday, April 29th by 6 pm

This homework is a **project** and is worth **15%** of your overall grade.

Your completed project will be due **Sunday, April 22nd, 2018 by 6:00 pm**. The minimum requirements for the project can be found below. Please look into it and come up with a good project idea. It can be anything ranging from an inventory management system, game, automating some process, etc.

If you have questions about anything in this document, please create a post on Piazza or come speak to an instructor.

Minimum Requirements:

- 3+ user defined classes
- 4+ data members per class
 - At least one class must include an array of objects from a class that you created.
- Appropriate methods for each class (including getters and setters)
- Your project implementation must include at least:
 - 4+ if / if-else statements
 - 4+ while loops
 - 4+ for loops
 - 2+ nested loops
 - 4+ strings variables/data members
 - File I/O for reading data members of an object
 - File I/O for writing to data members of an object
- Your project must have an interactive component (ask the user for input, create a menu for choices, ...).

Note: a project that meets the bare minimum requirements will only fetch you a maximum of 40 points out of 100. You can reference the Oregon Trail project to get an idea of how complex your project should be to receive points for going above and beyond.

What do you need to submit:

1. Checkpoint 1: Project Proposal (due Monday, March 19th by 6 pm)

Before you submit the project, you need to design and plan your implementation. You must supply a 1-2 page write-up for the project proposal addressing the following (you can always write more):

- What is your project?
- What is the goal of this project?
- How will you meet the minimum project requirements?
 - What classes do you plan to use?
 - Have you chosen their data members? How do you plan to use them?
 - What do you plan to read from a file?
 - What do you plan to write to a file?
 - How do you plan to interact with the user?
- Explain how the design meets the requirement.

You must submit your proposal to Moodle to get full credit for the assignment. The penalty for not submitting the Project Proposal is 10 points. You can write your proposal as a text file (.pdf) or as comments in a .cpp file. The Project Proposal is due Monday, March 19th by 6 pm.

Failure to submit the Project Proposal will result in a 5/100 penalty on your Project III score.

Your TA will review your proposal to make sure that you are not overly ambitious or not doing enough for this project. If the TA determines that your project needs to be altered, they will contact you for a meeting the week of March 19th.

2. Checkpoint 2: Class files & Code Skeleton (due Sunday, April 8th by 6 pm)

For this checkpoint, you will need to complete your class files and create your driver files.

Your .h files should be *complete* with all the data members and member functions you will be using for each class. For the class implementation .cpp files, you should fully implement simple functions like your getters and setters. For more complex functions you can include function stubs with detailed comments.

For example, if I were stubbing a function to implement bubble sort and return the number of swaps I might do:

```
/*
    1. Compare adjacent elements. If the first is greater than the
        second, swap them.
    2. Do this for each pair of adjacent elements, starting with
        the first two and ending with the last two. At this point
        the last element should be the greatest.
    3. Repeat the steps for all elements except the last one.
    4. Repeat again from the beginning for one less element each
        time, until there are no more pairs to compare.
*/
int bubble_sort(int arr[], int size)
{
    int swaps = 5;
    return swaps; // function returns expected type (int)
}
```

The driver file should contain detailed comments with pseudocode that explains the functionality of the project. (It should look similar to the libraryDriver.cpp you submitted for Homework 7 Part I).

You must submit your Class Files and Code Skeleton to Moodle to get full credit for the assignment. The penalty for not submitting the Class Files and Code Skeleton is 10 points. Zip all your .h and .cpp files and submit the archive file on Moodle by Sunday, April 8th by 6 pm.

Failure to submit the Class Files and Code Skeleton will result in a 10/100 penalty on your Project III score.

3. Checkpoint 3: Final Deliverables (due Sunday, April 22nd by 6 pm)

The final version of your project will be due on Sunday, April 22nd by 6 pm (no bonus available). You must submit a .zip file to Moodle which includes:

- All .h and .cpp files including the main driver program, correctly indented and commented.

4. Checkpoint 4: Project Report - Reflection Activity (due Sunday, April 29th by 6 pm)

Reflection questions:

- How did you prepare for the Project?
- Did you write a Code Skeleton? Was it useful? How?
- Reflect on how you could have done better, or how you could have completed the project faster or more efficiently.

What you need to do:

Write a 1-2 page report containing answers to the reflection questions.

In addition, write a paragraph answering the following question, in the context of the Project III in CSCI 1300:

Did you have any false starts, or begin down a path only to have to turn back when figuring out the strategy/algorithm for your Final Project program. Describe in detail what happened, for example, what specific decision led you to the false starts, or, if not, why do you think your work had progressed so smoothly, and give a specific example.

Note: all reflection papers should be individual.

Upload your report as a pdf on Moodle under *Final Project Report* before Sunday, April 29th by 6 pm.

Failure to submit the Report will result in a 5/100 penalty on your Project III score.

Collaboration:

All code written for this assignment must be your own. You may work together to come up with project ideas and to work through errors. Make sure to give credit to those you work with!

You may not use code provided or taken from anyone or anywhere else. **All code must be your own.** In particular, your projects should have different classes, different implementations, and different behavior.

If you use any resources (web or otherwise) you will need to acknowledge them at the top of your main program.

Example:

```
/*sources:
1.
https://stackoverflow.com/questions/9622163/save-plot-to-image-file-instead-of
-displaying-it-using-matplotlib
#used to find examples for how to save a plots vs showing a plots

2.
https://stackoverflow.com/questions/1557571/how-do-i-get-time-of-a-python-prog
rams-execution
#used to learn how to print execution time

3.
I worked with Jordan to brainstorm ideas for the project

4.
My TA helped my fix some of my errors with the function that writes to file
*/
```

You cannot copy all or most of your code from a resource, acknowledge it, and receive points for this project. If your code is very similar to an online resource or to one of your classmates or to one of the students who took this class (or another intro programming) before, it will receive a 0 and will guarantee you a meeting with the Honor Code, regardless if you acknowledge your source or not. You must show that you substantially contributed to the project, that your project does not contain just parts copied from other sources. If you have questions about what this means, please come speak to an instructor.

Interview grading for the project will begin on **April 23rd**, the Monday after the submission deadline.

Points:

If your code does not compile, you can get a maximum of 40 points for the project.

40 points for interview grading

- TA's questions about your project
- code compiles
- algorithms descriptions, comments, good style

40 points for minimum requirements (this is the *maximum* you can get if your code does not compile)

- code meets the minimum requirements specified on the first page (number of classes, loops, etc.)

20 points for going above and beyond

- creativity
- difficulty/complexity of project
- originality

Possible Deductions:

- 5 points for not submitting a proposal
- 10 points for not submitting code skeleton and class files
- 5 points for not submitting report