15-110 F26 - Final Project Reflection

Name:

AndrewID:

Complete the following problems in the fillable PDF, or print out the PDF, write your answers by hand, and scan the results. When you are finished, upload your **final-reflection.pdf** to **Final Project - Reflection** on Gradescope.

You are allowed to use generative AI on this assignment. However, you should still make sure you understand everything that is produced and can explain it in your own words.



#1 - Algorithms and Abstraction - 6pts
#2 - Fundamentals of Programming - 13pts
#3 - Modern CS Techniques - 12pts
#4 - Comparing Algorithms - 13pts
#5 - CS Ethics - 6pts

#1 - Algorithms and Abstraction - 6pts

Can attempt after Problem Decomposition lecture

How does your project use **problem decomposition** to make the code easier to understand? If your project does not use problem decomposition, how could you break down one of the functions into smaller parts?

#2 - Fundamentals of Programming - 13pts

Can attempt after Testing lecture

Choose a function in your project that is reasonably complex (at least a few lines long, uses at least two control structures). Describe what that function does in plain language - no code, just your own words.

Describe two tests you used to verify that your code was correct. One should be a test of an individual function, while the other should be a broader system-level test.

#3 - Modern CS Techniques - 12pts

Can attempt after Libraries, APIs, and Documentation lecture

What documentation and/or tutorials did you refer to while working on this project? What did you learn from them? If you didn't use any documentation/tutorials, how did you learn everything you needed to know for the project?

If you used generative AI tools to generate code for this project, reflect on your experience using these tools. If you did not use generative AI, reflect on your code-writing process.

In either case, what worked well, and where was there room for improvement?

#4 - Comparing Algorithms - 13pts

Can attempt after Tractability lecture

Choose a moderately complex function from your code (this can be the same function you used in the Fundamentals of Programming question). Ideate a reasonably different approach (not just surface-level changes) that could be used to solve the same problem and describe it here. (You can use either plain language or code.)

Compare your original algorithm to this new version. What are the strengths and weaknesses of each? Consider all the factors we discussed in the Comparing Algorithms unit - input restrictions, efficiency, and more.

#5 - CS Ethics - 6pts

Can attempt after Ethics of Environmental Impacts lecture

In lecture we've discussed a range of cases where programs had unintended consequences. Reflect on the project that you built. Could your project have unintended consequences? If yes, explain why; if no, explain why not.