CS 124 Spring 2023 **Team project**

Project Purpose, Plan and organization, Apr 22, 11:59pm (10 points)
Live demo/presentation May 9, 9:00 am (10 points)
Report Part 2, source, data files submissions May 10, 11:59pm (30 points)
Other assignments as posted (points vary)

Purpose

This assignment is to provide you with an opportunity to:

- practice software development project in a community-focused and/or socially-focused context
- practice/apply concepts learned in this course
- practice/apply your C++ knowledge and skills
- practice your collaboration skills

Requirements

For full credit:

- Project plan and organization (first section of report) which meets the requirements as described below and as listed in the rubric
- Remaining sections of report, which meets the requirements as described below and as listed in the rubric
- Source and data file(s)
- Live presentation by all team members
- Other assignments as posted

Description

Based on your ideas from brainstorming, select one idea to develop a **proof-of-concept** project using C++, executing in the Linux environment. Your team will define the operations needed for your program to make it useful to the users. For this, consider:

- What problem/issue does your project address? Provide the rationale for this issue with some research (1-2 credible resources, cited appropriately, can include materials your discussion post).
- Define the target users (those who may benefit), the scope of impact (household, neighborhood, regional, etc.)

Add discussion of the above in the Purpose section of your report.

As you develop your project, include these programming requirements:

- Use C++, execution in Linux environment, Ubuntu preferred
- Use specific functions which are no longer than 60 lines of code
- Apply course concepts from this semester including (must be listed and discussed in detail in comments and in report):

  - a linked list (no STL, use textbook starter code)
  - an array **or** vector (explain your choice)
  - At least one of: an O(n log n) sorting algorithm **or** a searching algorithm (no STL, use textbook starter code)
  - At least one of: stack **or** queue (no STL, use textbook starter code)
  - At least one of: set, map, priority queue
  - At least one of: hash table by separate chaining or by open addressing (no STL, use textbook starter code)
  - At least one of: binary tree (decision tree or BST) or graph, with traversal of the chosen data structure (no STL, use textbook starter code)
- Include comments to identify the programmer(s), date, and the project
- Include comments for any significant blocks/lines of the program, comments which explain the purpose of the code and not just what the code does
- Follow other programming practices/expectations as discussed in the class, including but not limited to
  - Executes in Linux environment, C++ 11 compiler with no extensions
  - Keep your functions concise and specific (60 lines or fewer)
  - Use separate compilation
  - No global "using namespace std" though you may include more specific sub namespaces inside functions. Text starter code must be adapted.
  - Keep your functions concise and specific (fewer than 70 lines)
  - Use local variables – global variables should be used only for constant values and this will be rare

Additional notes:

- If your program uses data, try to find data that is relevant and downloadable from a trustworthy source. If this is not possible, create your own sample data to show your program's capabilities. You may need to do both.
- Try to get first-hand perspective on the issue either with interviews of the target audience or by using your own knowledge if you have first-hand knowledge/experience of the issue.
- Ask at least 5 users (though more is better) to test your program and gather feedback, on the interface and on the program functionalities. Diverse representation of the target user group is also preferred if possible.

Optional Bonus (up to 3 points)

After the above requirements are completed,

- Include a graphical user interface (GUI). Other programming languages and IDE (must be widely available) may be used, but this part will not count toward the program requirements listed above. OR

- Include a hardware component to your project. Other programming languages and IDE (must be widely available) may be used, but this part will not count toward the program requirements listed above.

Report requirements

The project report include these required sections:

- Planning and organization
- Development process
- Product
- Pitfalls
- Possible improvements

The report should be submitted as a .pdf in Canvas per submission instructions. You may create the report in a document creation software of your choice. Please ensure that your report is cleanly formatted and free of distracting grammatical or other errors. These errors will cost you points on clarity. For more specifics, check Canvas for the grading rubric.

Purpose

Provide a few sentences describing the purpose of this assignment. Include your perspectives with original insights and thoughts on:

- What you are learning/practicing
- What your program does and the expected benefits. Include information to describe:
    - What problem/issue does your project address? Provide the rationale for this issue with some research (1-2 credible resources, cited appropriately).
    - Define the target users (those who may benefit), the scope of impact (household, neighborhood, regional, etc.)

Use your own words. (I don't need to or want to read my own words again.)

Rubric criteria: Provides evidence of a clear understanding of the purpose for this assignment (what you are learning) and this project (what your program does), including original observations and rationale. Rationale and research for issue, target users, expected impact scope are described in good detail.

Planning and organization

Describe the process you expect to follow for successful completion. This should be mainly in paragraph form, with list(s) of expected steps and visuals as needed for clarity. Use your own words. Though this does not need to include every detail, it should provide enough information so that your plan can be practically followed by another programmer of the same experience. (I will be reading it in this view.)

Include a timeline to show your teams' planned activities, what has been accomplished, and the expected timepoints for other activities to lead to the successful completion of your assignment.

At least one flowchart to show program flow is required. The flowchart must have a start and an end. All decision points should be represented as a diamond. Include sufficient details to describe all significant program operations.

A diagram to organize objects and interactions with other objects may be useful.

Listing of team members and the expected work division of the project should be included.

The plan can change through the development. Discuss any changes in the Development section.

Rubric criteria: Describes the process you will follow for successful completion of this project. Provides clear descriptions of the decisions made in structuring and organizing the project. Initial (possibly not final) answers to many of the following questions are provided: •What course concepts will be used? • What input is the program expected to use? Why?• What classes and variables will you need? Why? • What constants, if any, will you need? Why? • What functions will you develop and use? How did you decide what operations go into each function? • How will you decide that your program is complete? An effective timeline and a clear flowchart are included. Individual contributions are clear throughout. Details on how regular and frequent communication will take place and the logistics to ensure a successful collaboration are included.

Development process
Describe the process you actually followed in order to develop the code for this project. The following questions may be used to start your discussions. Overall, your entire development process should be described in sufficient detail that your process can be followed by another programmer of the same experience.

- How did you use the plan(s) you developed?
- Which parts did you choose to develop first? Why? Who did what parts? These questions should be addressed to describe the complete development process. This should be mainly in paragraph form, with lists of steps and visuals as needed. Use your own words. The description does not need to include every detail, but, as mentioned above, should provide enough details and information so that your planning process can be replicated.

- How did you test your program to make sure it works correctly?
- Be sure to discuss the development of all key segments of the program, specifically identifying those that help meet the listed assignment programming requirements.
- Discuss how feedback from users were collected, including how your team considered the feedback for possible program adjustments.
- Specific team member contributions should be clearly described.

Product (Video)

Submit a 2 to 3-minute (max) video showing your program execution. Plan this out carefully so that you are showing (off) your work as best as possible in the time allotted.

In the video and narration, show and describe the capabilities and limitations of the program with good test cases that cover expected and edge cases.

All team members are clearly present and contributing in the video.

Rubric criteria: Program executions are shown and described to show program operational requirements are met. Expected and edge cases are discussed knowledgeably. At least 2 test cases, with context of identified target users, included to show different capabilities and limitations. More test cases may be needed depending on program requirements. All team members presented as knowledgeable about the project.

Pitfalls

Describe any difficulties and issues you encountered during this assignment. Provide screenshots and other visuals as needed to describe these clearly. Describe how you resolved these issues and what you learned through them.

If somehow, you didn't have any difficulties, issues, programming errors, or bugs, include an explanation of why this could be the case. This is VERY rare and we will most likely need to meet with you for more clarifications.

Rubric criteria: Provide clear descriptions of any issues, errors, challenges encountered in this assignment. Describe how these were resolved and what you learned from them. If working with a partner, discuss how collaboration may have affected these pitfalls and how you resolved them.

Possible improvements

Based on user feedback and your own analysis, include specific and constructive discussions on the following:

- What could you have done differently to improve your completion of this assignment? If working with a partner, include how your collaboration could have been more effective.
- What specific action will you take to improve completion of your future assignments? How could your team collaboration could be more effective for long-term team success?
- What could be improved about this assignment overall for you and/or for future students to gain experience and show mastery of the concepts?

Rubric criteria: Clear and specific discussions on the following: •What could you have done differently to improve your specific completion of this assignment? If working with a partner, include how your collaboration could have been more effective •What could have been done differently for better completion of future assignments? How could your team collaboration could be more effective for long-term team success? •What could be improved about this assignment overall, for you (or other students) to better demonstrate a successful and mindful mastery of the programming concepts covered so far?