CSC 2220 – Programming in Java
Fall 2024 Semester, Projects 4-5
Due Date(s):
Project 4 – 11/22/24, by 11:55pm Sharp
Project 5 – 12/6/24, by 11:55pm Sharp
Note: Group Assignment

Problem Description

We have been plagued with an online degree audit system that isn't quite correct. With massive changes to the Liberal Arts Core that began in Fall 2024 for incoming students, now seems like a good time to create a tool (utilizing Java) for us to create one.

In order to ensure the success of this project, everyone will be working together to come up with a unified look-and-feel, as well as functionality, that will assist in advising students in upcoming semesters and years. I have included a pdf that contains the proposed structure of the application.

Ultimately, student information would be saved in text files, then brought back up when time to get advised for the next semester, and update accordingly.

Guidelines

- To say it needs to be easy to use is implied; perhaps a better way to think of addressing this tool is to ask if it is useful.
- Since we're utilizing screens, make the font bigger than what you would use if you were writing a paper (maybe start at 14 point and see if that is sufficient?). As you may/not have heard, the resolution of printed text is far higher than anything generated for the screen.
- I am open to alternatives that lead to fewer mouse clicks. This is what makes Canvas the bane of my existence.
- That being said, the checkboxes for the majors is non-negotiable. I plan on extending this project out beyond this class, and that would be some of the future work that would be necessary.
- I am happy with the color scheme, but go with colors that are lighter on the eyes, like maybe a pale green, pale yellow, etc. Anything with a color to it is the background color, not the color of the text. The color of the text should remain the default color. Would bold help? Maybe something to look into…

- Would an application with a traditional menu structure be a better route to consider? Is there something else I didn't think of? I am open to changes, just as long as you tell me you are making whatever change and what your rationale is behind it. I may provide guidance/feedback. At that point, you'll have the final say. Just expect some critiques…

Student Record File Format*
Prefix Number Title Hours, one per line. So for someone in CS:
CSC 1180 Programming in C++ 4
CSC 2180 Data Structures 4
*might be useful to have one more piece/string at the very end for completed, in progress, or registered.

From what we saw last week, we can pick off what we need to via StringBuilder.

Due Dates for Projects

- **Project 4 – Due 11/22/24, by 11:55pm Sharp –** Get the GUI side of things operational. This means that buttons lead to the appropriate screens popping up, while others go away. You may also choose to start working on the back-end logic of the application.

- **Project 5 – Due 12/6/24, by 11:55pm Sharp –** A finished application. This means that your application will correctly display a student's progress. You will also need to either have a file to read from (if the student already has a history), or create a new file (newly-assigned student). These files will need to be updated/saved for the next advising session, or if students need to come by and ask questions about their progress. The file will need to somehow keep track of whether the student has finished, is in progress, or has registered for a class, then allow for some update (like, all/some classes in progress were completed with a decent grade; signed up classes are now in progress, where appropriate).

Miscellaneous
- Each of the files that you will have at your disposal addresses a certain component. One will be for the liberal arts core, four will be for each area of contextual learning, then a major core, then a

major's electives. At this point, Computer Science and Software Engineering have been included. I need to do MIS, and will update the zip file when I have gotten that uploaded.
- Assume the first Catalog Year with this application will be the 2024-25 school year.
- It might be good to create a driver, then add the GUI classes separately through WindowBuilder. Theoretically, this should alleviate creating each GUI, then having to figure out what to do to make all GUIs work together and play nice, especially if each of the GUIs has a main().
- On the back end, whatever file is created should then check against requirements for major, liberal arts core, etc. The order of completion should not matter. If Java, for instance, is found at the bottom as completed, it should show up in CS Electives as the first item, like how it's currently listed.

Schematics Explanation

- Start Screen (1)
    - Provides four options; hitting any of the four will lead to the second screen. You should denote the difference in what functionality to be completed via the title bar at the top of the window.
    - All four options are buttons.
- Input Screen (2)
    - Labels and Textfields to enter first name, last name, and catalog year. (maybe a combobox for catalog year?)
    - Checkboxes for the four majors
        - Again, assume only one major selected
    - Next Button leads to the next screen
        - Update Student Record – Screen 7
        - Create Student Record – Screen 8
        - Change Major – Screen 3
        - Delete Student – Screen 5
- Change Major (3)
    - Checkboxes for the different majors. User has to hit "Next" button to confirm the change.
    - Label at the top shows current major.
    - Next leads to Screen 4
- Major Changed (4)

    - Message stating major has changed.
    - Clicking OK goes back to Screen 1
- Delete Student File Confirmation (5)
    - Label to ask – “Are you sure?”
    - Buttons
        - Yes – Screen 6
        - No – Screen 1
- File Deleted Confirmed (6)
    - Text to say student file deleted.
    - Button at the bottom says, OK
    - Pressing button goes back to Screen 1
- Student’s Current Progress (7)
    - Load student file, and check to see what has been completed. Provide buttons that allow student to see what they’ve done/where they currently stand.
    - Next Button – add courses (Screen 8)
- New Student Progress (8)
    - Same as 7, but all options are red.
    - Next Leads to Screen 9
- Completed Courses, 1 per line (9)
    - Text Area to add courses one per line. Doesn’t matter at this point if new/returning student.
    - Next Leads to Screen 10
- In Progress (10)
    - One per line, as well
    - Next Leads to Screen 11
- Registered For (11)
    - One per line
    - Next Leads to Save Changes
- Save Changes (12)
    - Buttons to save changes
        - Yes – Changes confirmed
        - No – Back to Screen 1
- Clicking on Buttons from 7 will show a summary of courses taken in a given field.
- Need to figure out a strategy to update courses in progress, registered courses.

Files Included

Contextual Coursework – Each of the following has courses that count towards each of the four components, along with hours for each course:

- contextualCourseworkCommunity.txt – for the community component of the 21 hours needed. A student needs at least three hours to fulfill this.
- contextualCourseworkNation.txt – same as above. At least three hours needed.
- ContextualCourseWorkSelf.txt – same as above.
- ContextualCourseWorkWorld.txt – same as above

csCore.txt – Computer Science core classes for the CS degree
csElectives.txt – Computer Science electives for the CS degree
csMathScienceRequirements.txt – Math and Science Requirements for the CS degree

humanitiesFineArts.txt – Students need to have three hours from certain three-letter prefixes to satisfy this. The prefixes are listed in this file.
socialBehavioralSciences.txt – Same note as with the humanities file. Three hours are needed from certain prefixes within the contextual coursework applied.

seCore.txt – Software Engineering Core
seElectives.txt – Software Engineering Electives
seMathElectives.txt – Possible courses in math SE majors could take

liberalArtsCore_2024.txt – the list of classes possible. Seminar is needed (1 hour), 2 Science courses with labs, 2 Math courses, 2 Composition courses, etc.

Last Notes

- As you progress in this project, I don't expect you to have all of the answers, as I'm seeing some things that I don't address**. If you have questions – ASK. I'm not the temperamental type to get mad at questions. There's probably some detail I've left out or made unclear.**
- Make sure you're pulling your fair share of the work. I know group work can suck if someone's not holding up their end of the bargain. And I will be checking…
- I'll try and craft something along the way, in case you run into issues, I might be able to provide a solution, or rather, direct you towards a solution.
- Filenames need to be in the following format: LastName_FirstName_Major.txt
- I welcome any ideas for future work at the end of the project.
- Only one submission of each project is needed by the appropriate due date/time. I don't care who uploads the work, just that it's there.