HANRÈ (WPJ) DELPORT

50605216

CMPG 121 CREATIVE PROJECT: PHASE 2 – REFLECTION

1. REFLECTION ON AI USSAGE

During both the planning and developing phase of this project I limited myself on the use of AI.

Planning Phase:
In the planning phase I used ChatGPT to generate some creative names for my educational platform, but none of them stood out to me. I tried to combine some of the names and finally decided on “SkillRise”.
I also used the QuillBot add-in provided by Microsoft word to get some synonyms for words that did not make sense in my sentences. Furthermore, I used it to change the construction of specific sentences with faulty wording. This helped me expand my vocabulary and at the end of the planning phase I did not need to use this tool to help me write clear sentences.

Developing Phase:
I used ChatGPT to generate a list of 50 user names, surnames, unique 6-digit codes and 6 random marks out of a 100 that I used in my “Users.txt” file as test data. I also had ChatGPT generate course names and descriptions that an educational platform like SkillRise would offer to students.
I used ChatGPT on some occasions when debugging my code, especially when working with classes and objects because of my limited knowledge on OOP with c++. However, after the second or third time struggling with a bug, I started to understand how to better incorporate classes and objects.

2. REFLECTION ON ADHERENCE TO TECHNICAL AND ADVANCED REQUIREMENTS

My program met all the technical requirements that we were given at the beginning of this project:

- I used three arrays. Two “User” arrays, one to store data regarding up to 60 different students and the other to store data of up to 10 Lecturers (objects of the User Class). I used a “Course” array to store data of up to 10 courses (objects of the Course Class). Through the whole program I used other arrays to store the marks of students and the averages of students with the same course, etc.
- I did not have to use structs because I already implemented Classes.
- I defined functions to simplify the programming process and to divide my program up into different smaller sections. I wrote functions to display all the different courses that SkillRise offers and another function to display the marks of a student together with the weights that each assessment counts towards final 100%. I defined a function to display all the statistics of a specific course, it receives course object and an array of users then run through the users to check who attends the course and to calculate the mean, min and max mark etc for the course. Other functions I used included functions to edit marks of students, to save new students to the text files, to read data from text files into arrays etc.
- My program is user-friendly with clear and formatted output. User input is standardised and clear explanations is provided regarding the expected format for specific user input.
- I used conditional statements and loops for different tasks such as running the menu as long as user does not exit or checking if a student is registered by running through the array of registered students. I also used loops to sort an array of averages and to generate a 6 digit number until it is unique and even.
- I used two text files. One for permanently storing the data of all registered users and another for storing the data of all courses. The “Users.txt” file stores the user’s name, surname, generated ID ,course signed up for (if applicable), marks for each assessment and status of course (if the user finished the course or not) separated by spaces so it can be extracted and used to create objects in the main program. Similarly, the “Courses.txt” file includes the course’s name, a description of the course, the course’s ID and the weights of each assessment separated by the #-symbol and a new-line character (to separate weights from other data).
- I used error traps (in the form of do-while loops) to do input validation and to ensure that no faulty input is accepted.

My program also met some advanced requirements that was listed:

- I defined 2 classes. A “User” class to store all the needed information of the users, such as their name, surname, course signed up for (from the Course class) and unique, random generated student-ID. The other class, “Course” Class, saved the course information, such as the course’s name, a description of the course and an array of the weights that each assessment counts toward final mark.

The “User” class also define some methods for the initialization of the objects and methods to extract data from the object or to save data to the object.I defined a GenerateID() method to generate a random 6-digit number and a CalcAverage() method to calculate the average of a student by multiplying their marks for an assessment with the weight of the assessment, etc. I only defined two methods inside the “Course” class, both were constructor methods, the one with no parameters and another with three parameters (the name, description and weights of the assessments of the course). Unfortunately I did not put enough time away to incorporate smart pointers into my program.

3. OVERALL REFLECTION ON THE PROJECT

I did thorough research in my phase 1, including pseudocode for different functions and some flowcharts explaining the main- and supporting functions of my program. These tools definitely helped me in the developing stage, giving me a clear picture of what needs to be done and how it will be done.

I had to remove some of the functionalities that I initially intended to have in the program because I did not want to repeat functions and functionalities to much. For example, I planned to have the user view courses, but this was unnecessary because each student gets a chance to view courses when they first sign in and the lecturer also is able to view different courses.

There were many unforeseen problems and challenges that I had to overcome by defining new functions that I didn't initially realize would be needed. Many of the pseudocode that I wrote did not always work as expected and it had to be a little altered. For example, with the EditMarks() function, I didn't realize that I won't be able to find the user's position in the text file and rewrite their entry only, so instead I had to write a whole new function that clears the text file and then writes the data of the two arrays to the file.

I enjoyed the creative process of identifying a real-world problem and doing research to solve the problem by using the programming and problem solving techniques we learned in this semester. I also learned that not everything will always go according to plan, so you should always be flexible in the way you work around and think about a problem.