Background

Looking back, the main objective of creating the application being to make an app that that will be relevant to my career, and something that I will work more towards in the future. The ideas of what I should create were almost infinite, But the one thing that kept back to me was an application that would help me study better for upcoming tests and quizzes. And taking Quizlet as an inspiration, I've decided to make a flashcard app for as of now would contain general functions that would be useful for just the creation of the question-and-answer pairs and then studying based on them.

C++ concepts Used

Recursion: The input function in the program uses recursion in order to store the questions and the answers that are entered by the user within the **<Flashcard>** vector. The count number is the number of times the recursion occurs, for instance if the number of questions-and-answer pairs the user needs to input is 3 then the code will run at count = 3, then count = 3-1, and then count = 2-1, and after that when count = 0 function will end.

Classes: while the code does not contain any inheritance due to not having any child classes, it does focus on the use of the main class Flashcard in order to make the entire program function. The Flashcard class contains all the necessary main functions to retrieve data and output the necessary data. But I do have future plans for adding base classes when I add categories to the applications.

Data Structures(Vectors): The use of vector was the most optimal when it came to the application that I was building, at first, I started using lists with a limited number of values that can be inputted, then I switched to the use of vectors. While it was hard in the beginning due to the advanced code that was needed to complete the program, it ran the program the smoothest. Especially due to the fact that the user is able to create an infinite number of flashcards.

Streams: The program uses file input and output and also uses user interaction in order to get data. Such is seen as the user enters the question and the answer pair, which gets stored in a file for future use. Then the program retrieves the questions and answers from their dedicated file in order to output for the user to run the flashcard session.

Challenges and Changes in decisions

The major challenge I faced was multiple errors which occurred in outputting the data taken from each of the files that stored the questions and the answers in a random order. Initially instead of having its own function, I tried randomizing the question and the answers in the quiz function where the individual pairs of questions are shuffled instead of outputting the pairs in the order, they entered it in. The problem I faced was that the questions and answers were not matching. And a random answer would be paired with question. With the use of the function, I was able to test each problem I faced separately rather than having to run the program every time. And after fixing it I was able to use that function in my application.

Another major error was the requirement to hit the ENTER key twice after entering the answer to move onto entering the next question, which took a bit of help from my friends and constant search on the web, not even ChatGPT was helpful in providing the right feedback.

The change in the use of classes was also helpful in keeping track of what functions can be used to gain access to the flashcards. The initial use of struct had its benefits, but the use of a class allowed for a more organized look and easy access when it comes to updating the program in the future.