Project 2 Documentation

Mackenzie Zappe

Lab Section 1105

The purpose of the program is to act as a car dealership database, in which I take in a list of the cars from a file, sort them by ascending price and print to both the terminal and another external file. I can then use these processes to have the user ask about the available cars price and display that. Finally, I can have the user rent a car with the data needed.

The design of my code is based around modularity. I first took the purpose of the program and then divided it into different objectives. The first objective was to read in the list of cars. I then decided that this objective had only one step and therefore only one function needed to accomplish it, so I implemented a readCars() function. The next objective was to print the list to the terminal which again only needed one function to finish, so I implemented a printCarsTerminal() function. The next objective was to print to an external file. Like the printing to terminal objective, this is only one step and furthermore the same structure. Next I had to sort the cars from the file by price. For this module, I decided I needed to break it down further to accommodate its complexity. The sub-objectives included a swapping mechanism and a sorting mechanism. The swapping mechanism yielded a temp variable to store the swapping value. The sorting mechanism yielded a selection sort function, combining the swapping mechanism with price specific ordering. Next I had the rentalCarQuery(). Here, I calculated the price for the available cars based on the user’s inputted amount of days. I made sure to check if the cars availability was true before I displayed them to the screen, which I used a similar mechanic to the printCarsTerminal() function. Finally, I had to create a function to allow the user to rent a car. I had to check if the user’s choice was available, calculate the rate for days they wanted it for, and change the availability if rental is possible. After all this I implemented a printing mechanism.

My biggest problem was adjusting to the syntax of C++ and its different structuring of the I/O stream. This slowed me down as I had to check the documentation over and over to ensure I was implementing correctly. Design-wise I had no major problems. The process of the selection sort was a little tedious to work out but overall made my code easier to implement. Also, I struggled finding the .h extension when saving my header files so, initially I had written my header files in a .cpp source file which did not allow the actual .cpp files to use my header files. Eventually, I figured out how to implement the extension so the header files could link to the source files