Documentation
Project #2

Project number 2 is about learning how to program arrays of structs while implementing a car rental program. The main objective for the program is to allow functionality to the user to select from a menu that gives them the option to read from a file filled with cars. The user will need to make sure that whatever file name they enter it needs to include .txt as an extension. If the user entered the correct input, the program continues, otherwise, they are instructed to enter the file title again. Once the title is correctly entered, the program will read in all of the information from the file. If there isn't any information, the program exits. The program than clears the screen for more easily readability.

The newly created struct is now passed into the sortAscending function which will automatically sort the cars in price ascending order. If there isn't any data, or if the file that we're reading from is corrupted the program exits. We then enter the file writing phase which will write the contents of the array of structs into the file specified by the user when prompted. A display message will be issued to the user and verify the file was written. If the program encountered an error while opening the file, the program exits. If the program encounters any other errors, it exits.

Then, clear the screen for readability and we'll print the newly sorted array of structs out to the terminal. You'll be able to inspect this inventory list and verify that it is in the correct order. The program will then ask the user how many days they would like to rent a car for. The program clears the screen and we enter the carSelection function. Once inside of the carSelection function the availableCars function is used to display only those cars that are actually rentable. The numbers listed for the car represent the number of the car and it is important that the user enters numbers that are written exactly like they are displayed next to the car. For example, if a car is labeled "car 2", then when prompted, enter in the number exactly as it appears. The program will deal with sorting out the correct index number. If the user entered a value out of the range or if the user enters a number that represents an unavailable car, then the program displays an error message and the program exits.

Upon selection of a valid car and the screen will let you know that you've successfully rented a car and display your total price and let you know of the next steps associated with renting. An exit message is displayed and the program finally exits. This concludes the program and all of the requirements. Additionally, I've included the mystrcmp and mystrcpy functions. You'll see all necessary documentation above the function in the comments, they will clearly explain what they do. There are three files that are included with this program a .h and 2 .cpp files. I've decided to make a library of all the functions I write in this class. If I don't use them, it's documented.