Kripash Shrestha

CS 202 Spring 2017

Project 2 Documentation

2/8/2017

The purpose of the program is to use the functions user_enter, read_file, print_data, estimate_rental_cost, most_expensive_car and print_available cars with an array of structs to open a data file with 10 different cars and copy it into an array of structs. The user will also be prompted to enter the name of the file and make several other decisions from a menu which makes the program either read all the data from the file, print out all data for all of the cars, estimate car rental cost, find the most expensive car, print out only the available cars and exit the program.

My design/ thought process for this was to use the exact file stream that I used in project 1. Except, this time it would be in a function and I would just call that function. This was not so hard because it was very similar to the first project. The only different thing was copying the data from the file to the strut members. I just had to specify with the dot operator in this case. I worked on all of my functions at first as a way to divide and conqueror up the problem as mentioned in class. All of my functions were pretty straight forward. I used a function to get the menu choice of the user and passed an integer value by reference. I used a function o read the file and copy the data of 10 cars as mentioned above. I also used a function to print the data of all of the cars, which included using a for loop that iterated 10 times and printed the members of each members of the iteration to the terminal. I also used a function to estimate the rental cost of the car by prompting the user to enter a number that represents the car. The function would then find

the cars location in the array of structs and then go to its price member and multiple it by the number of days that the user wished to rent the car. To find the most available car, I used a function that iterated 10 times, compared the values of each cars price and found the smallest value and stored it in an integer. Last of all I used a function to print available cars by using a for loop that iterated 10 times to check the availability of each car. An if statement within that for loop would check if the car was available, and then print the data of all of the available cars to the terminal.

The functions were probably the hardest part of the program. Now all I had to do what either use a while statement or do while statement with a switch statement to properly execute the program. This was not necessarily hard but tricky to do. I tried to use a while loop with for and if statements inside of it for each menu choice but it did not work and I ran into multiple bugs which included the program kept asking for the user to input their menu selection. It just kept checking that in an infinite loop even though I had a break statement in it. However, with that while loop my program did end when I enter 6 so there was probably something wrong with my jumbled up for and if statements for each menu selection within that while loop. So I looked back to a similar project with a user interactive menu we had to do for CS 135 and saw that I used a do-while with a switch statement inside. I tried that out and switched the menu selection that the user enters and then called a function for each case. And of course the program would exit when the integer 6 was entered by the user and I made a default that sent a terminal notification to the user to tell them to enter a number between 1 and 6.

If I could make a change if I was given more time, it would have to be with my implementation of two functions for two different menu selection choices when I could have easily used once. I could use a function with two parameters, one being the array of structs and

two being the bool flag. I would use a function with a for loop that printed the data for either all of the cars or for the available cars based on the bool flag. For the print all of the data, I would pass the parameter to check for True or False and would therefore print all of the data for the cars since they can only contain 0 or 1 in the cars' bool member. For print available car data, I would pass the parameter with the bool to check true only and would therefore print the data for the cars that are available / contain 1 in the car's bool member. One other thing that I would definitely work on is the visual representation on the terminal. I noticed that someone had mentioned a couple of libraries for C++ for print formatting on the discussion board on web campus. If given more time, I would definitely take a look at the library and look at some functions that I could use to make the print formatting to the terminal aesthetically pleasing. Right now it is just really weird and a little difficult to read when they are just printed as represented on the file.