Kripash Shrestha

CS 202 Spring 2017

Project 4 Documentation

3/2/2017

The purpose of this program is copy the a datafile with of 1 agency which has 5 cars and up to 3 sensors and a possible owner. This will be done with the manipulation and data encapsulation of 3 different classes, CarSensors, RentalCars and RentalAgencies. This program will be constructed with while loops, arrays, arrow operators, pointer arithmetic, my own string compare and string copy functions and data encapsulation. One of my first approaches to finishing this project was to separate the classes into their own class.cpp files and the class.h files and include them to each other. However, as I was nearing the end of my project, the compiled through errors with undefined references to the functions and classes so I decided to combine it into one giant header file and my main cpp file. One of the things to keep in mind is that the following program contains functionalities of the previous project and this so the functionalities are Read all data from file, print all data from file, estimate car rental cost, find the most expensive car, print out available cars, print out total sensors in agency and find the most expensive available cars.

One of the most difficult parts of this project for me was the getting rid of the "}" curly brackets in the sensor part of the file stream. I tried to take the entire thing as one string until it reached the "}" bracket and then separate the sensors into 3 different strings based on white spaces. However, that gave me a lot of problems when I started running into segmentation faults.

At this point I decided to get the individual sensors from the file stream. The breaks for this loop that I used would be the "}". That was one of the conditions I used. The second condition was the white space, and a white space would determine if to break the loop and go onto the next sensor or not. This was not by pointing to the file stream, an array inside my function and a pointer to the pointer file stream. This was one of the hardest parts and one of the most essential parts of the project because this is required for me to finish implementing the other parts of my project. For example, this is required for the extra cost of the sensors. This was determined by pointer arithmetic, reserved strings and a compare string function. I used the compare string function to determine the extra cost that would be added to the base price of the sensor based on the return value of the compare string function for 5 different outcomes. One outcome for each sensor and then another outcome for no sensor. This was a gate way to my project and I was able to finish my file stream copy and get information for my other functionalities, such as finding the amount of car sensors in the agency by sensor type. One thing to keep in mind is that the sensors counts are stored into static integers so every time I read a different file, I have to reset all of the static integers to 0.

Another thing that I struggled on in this project was with the find most expensive and available car functionality. The asking if the user wanted to rent or not and then storing it into the car's owner member was the easy part. To find the most expensive available car, I recycled my find the most expensive car functionality code and added a second condition to where only the **FINAL PRICE** of available cars would be considered. So, I tried using pointer arithmetic with the two conditions in one if statement but that did not work and it resulted in a segmentation fault because I realized that for this to work the first car has to always be an available car. So to fix

this I incremented the first pointer until it found a car that was available. I had a counter increment every time the while loop iterated to make sure that I did not run into segmentation fault when iterating the second for loop. I then pointing my second pointer to where the first pointer was. I then incremented the first pointer and the counter once more. Then I ran a for loop from the counter to 5, since 5 is the maximum Then I ran the second for loop, which contains an if statement that checks to see if the car that the first pointer points to is both available and large than the final price of the car that the second pointer points to. If that is so, then the second pointer will pointer to where the first pointer is pointing. This is continued until the for loop stops. The car that the second pointer points to will be the most expensive available car in the agency fleet.

As mentioned above, I had originally planned on submitting a cpp file and header file for each class and a main source and main header file, but that was quickly shutdown with compilation errors I was getting. So, if I were to redo this project, or had more time, I would definitely learn about make files and do it the way I had originally planned to do so. That was it makes it easier for people to look and search through my code, rather than looking through hundreds of lines on a single file. Also another thing that I have yet to change is the fact that I am still prompting the user to enter 1-5 for the car that the user wishes to rent. If I had more time to do this project, I would change that to make sure that user enters a string either based on the model or make of the car and use string compare to find the car that the user wishes to rent and do my multiplication based on that.