Computer Science Project 2 Report

1. ***Obstacles during Project 2***

Obstacle 1: On my first draft of the code that I wrote, I used the figures given on the Project Spec page to test whether my logic was correct. I did this before I introduced the lines of code that would terminate the program and inform the user if they typed any data incorrectly. I noticed that upon entering the name ‘Lewis Hamilton’ into the ‘Customer name:’ section, the program would print out the next two *cout* statementswithout waiting for an input from the user and terminate. So, I tried a few different names to see if it would work eventually. What I noticed was that the program would run successfully producing an accurate output when there was a single name used for Customer name. When it was 2 names (or a full name), it would not work. After doing some digging around and finding the ‘*Some things about Strings’* hyperlink, I read that *cin* would stop the input as soon as it detected a space. To overcome this issue, I used the *getline* command. I then ran the code again with the name ‘Lewis Hamilton’ and it ran successfully.

Obstacle 2: While doing some finishing touches on my code, I noticed that we needed to give our output up to two decimal places. In the few calculations I had tried prior, the output was always a decimal number (non-integer) and it printed with two decimal places. But I was curious to see whether a whole number output would also have two decimal places. On entering a few values, I received the output $267 with no decimal places. To fix this I tried floating the final *totalCost* to check whether that would result in a correct output, but it did not. After a few minutes of reading, I came across the precision function that would allow you to specify to how many digits you want your arithmetic outputs to be. Using this, I was able to correct the issue that I observed with decimal places.

1. ***List of Test Data***

Fortunately, I was able to finish the program on time and it handled all the test cases in the way that the spec required.

The following are the lists of test data that I used to test each output. It will be in the order:

**(odometerStart, odometerEnd, rentalDays, customerName, luxuryCarStatus, monthNumber).**

If it’s not included, it means that the program should have terminated before reaching that point where the input is asked for. All 9 sets of data were handled correctly by my program.

1. Negative start odometer reading: (-1)
2. Final odometer reading greater than start reading: (42, 26)
3. Inputting negative for the number of rental days: (42, 52, -43)
4. Blank customer name: (42, 52, 2, ‘’)
5. Giving a random input for a yes/no question: (42, 52, 2, ‘James Hunt’, ‘a’)
6. Month number that is beyond 12: (42, 52, 2, ‘James Hunt’, ‘y’, 14)
7. Multiple sets of test data that I used to ensure that computation/logic is correct:
   1. To see if whole number output has two decimal places: (10, 110, 10, ‘Lewis Hamilton’, ‘n’, 10). Expected output: $457.00
   2. Regular data to check logic: (2417, 2754, 4, ‘Lewis Hamilton’, ‘n’, 10). Expected output: $248.77
   3. Regular data to check logic: (1885, 1973, 1, ‘Richard Petty’, ‘y’, 2). Expected output: $94.76