CS 31 Lecture 1 – Professor Smallberg Leo Gretzinger

Project 2 Report 10/15/18

**Obstacles:**

I faced three main obstacles when creating the program for Project 2. The first came when deciding which conditions would constitute an error after the Luxury question was answered by the user. First I had to figure out the syntax of an if condition with ‘does not equal’ or ‘!=’ signs, and whether or not ‘y’ and ‘n’ should be compared as characters or strings. When I determined that comparing them as strings would work better than as characters, figuring out if there should have been an && or an || operator between the two statements took a couple times to get right. Also, the first few times after I thought I had successfully completed the program, the output was a rental charge with no decimals, even though I included the “cout.setf(ios::fixed);” and “cout.precision(2);” lines. I struggled with this slightly, but then found out it was the fact that I had initialized rental charge as an ‘int’ instead of a ‘double.’ Lastly, the large amount of nested if-else statements at the end, necessary for calculating the final rental charge, was complex and challenging. To solve this, I went to slowly, step by step and made sure to account for every condition mentioned in the Project spec and ended up with an excellent output!

**Test Cases:**

For my test cases, I checked every range of miles, including boundary cases, both luxury and basic and both winter and non-winter months. I also checked if the error messages worked for each input.

The inputs are in the order: (starting miles, ending miles, rental days, customer name, luxury car, month)

* Luxury car; travelled exactly 0 miles 🡪 (0, 0, 1, Dude, y, 1)
* Basic car; travelled exactly 0 miles 🡪 (0, 0, 1, Dude, n, 1)
* Luxury car; travelled under 100 miles 🡪 (0, 45, 1, Dude, y, 1)
* Basic car; travelled under 100 miles 🡪 (0, 45, 1, Dude, n, 1)
* Luxury car; travelled exactly 100 miles 🡪 (0, 100, 1, Dude, y, 1)
* Basic car; travelled exactly 100 miles 🡪 (0, 100, 1, Dude, n, 1)
* Luxury car; rented in the winter; travelled between 100 and 400 miles 🡪 (0, 300, 1, Dude, y, 1)
* Basic car; rented in the winter; travelled between 100 and 400 miles 🡪 (0, 300, 1, Dude, n, 1)
* Luxury car; rented in non-winter; travelled between 100 and 400 miles 🡪 (0, 300, 1, Dude, y, 6)
* Basic car; rented in non-winter; travelled between 100 and 400 miles 🡪 (0, 300, 1, Dude, n, 6)
* Luxury car; rented in the winter; travelled exactly 400 miles 🡪 (0, 400, 1, Dude, y, 1)
* Basic car; rented in the winter; travelled exactly 400 miles 🡪 (0, 400, 1, Dude, n, 1)
* Luxury car; rented in non-winter; travelled exactly 400 miles 🡪 (0, 400, 1, Dude, y, 6)
* Basic car; rented in non-winter; travelled exactly 400 miles 🡪 (0, 400, 1, Dude, n, 6)
* Luxury car; rented in the winter; travelled over 400 miles 🡪 (0, 450, 1, Dude, y, 1)
* Basic car; rented in the winter; travelled over 400 miles 🡪 (0, 450, 1, Dude, n, 1)
* Luxury car; rented in non-winter; travelled over 400 miles 🡪 (0, 450, 1, Dude, y, 6)
* Basic car; rented in non-winter; travelled over 400 miles 🡪 (0, 450, 1, Dude, n, 6)
* Negative starting miles error message 🡪 (-1); program ended with correct error message
* Ending miles greater than starting miles error message 🡪 (3,2); program ended with correct error message
* Negative rental days error message 🡪 (1, 2, -1); program ended with correct error message
* Blank customer name error message 🡪 (1, 2, 3,); program ended with correct error message
* Letters other than ‘y’ or ‘n’ error message 🡪 (1, 2, 3, Q); program ended with correct error message
* Month not from 1 to 12 error message 🡪 (1, 2, 3, Dude, y, -12); program ended with correct error message