Ryan Lam

**Notable Obstacles:**

One notable obstacle was retrieving a string input from the user after a number input. Like we discussed in class, the extra newline character left behind after the number input was taken is read by the next getline function for the string input, thus skipping that input. To solve this problem, I implemented the ignore function, cin.ignore(10000, ‘\n’), to pass over the leftover newline character and take in the correct string input.

Another notable obstacle was the process of calculating total price based on the amount of miles driven. Translating the equations for determining rental charge into code oftentimes lead to error as I overlooked some condition or incorrectly calculated the amount of miles to charge for.

**Test Data:**

Negative starting odometer value: (-5, 500, 3, “Ryan”, “y”, 7)

Final odometer value smaller than initial odometer value: (450, 250, 2, “Ryan”, “n”, 3)

Negative value for number of rental days: (100, 300, -9, “Ryan”, “y”, 10)

No input/empty string for customer name: (500, 1000, 7, “”, “n”, 8)

Input for “luxury” condition neither “y” nor “n”: (5140, 5532, 6, “Ryan”, “”, 4)

Month value out of 1-12 range: (603, 780, 2, “Ryan”, “y”, 45)

Very large odometer values: (10000, 60000, 5, “Ryan”, “n”, 9)

Very large rental days values: (56, 84, 5000, “Ryan”, “y”, 5)

Very large odometer and rental days values: (20000, 50000, 7500, “Ryan”, “n”, 11)