Project 2 Report:

1. A brief description of notable obstacles you overcame

The notable obstacles I had in project 2 was the process of calculating the charge per mile driven when the mile went over 100 miles but not no more than 400 miles. In this phase, the calculating process became a bit more complicated because the mile rate was a dependent of the season the car was driven, furthermore we also need to handle the mileage cost of the first 100 miles.

My solution to the problem was first draw a time line and break the total cost into multiple cases such that

***Total cost* = *Base charged* + *Mileage charged***

and I break ***Mileage charged*** into further cases:

before going into the calculation process, I first determine the season mile rate, after the user input the season the car was driven, so I wouldn’t have to do too much of nesting in the if else statement below.

When the total mileage is less than or equal 100 miles, I just simply calculated the cost by multiplying the MILE\_RATE to the total mileage

if ( totalMileage <= 100 )

costMileage = MILE\_RATE\_100 \* totalMileage;

When the total miles went over 100 miles and less than or equal to 400 miles, I just calculate the cost of first 100 miles by multiplying the rate to 100, after that I subtract 100 from the total mileages and calculate the remain mileage cost by multiplying the seasonMileRate I found above

else if ( totalMileage <= 400)

{

costMileage = MILE\_RATE\_100\*100;

costMileage += seasonMileRate\*(totalMileage - 100);

}

1. Test cases :

// Error Message test cases

* 1. Start the Odometer with negative number – test if error message show up
  2. Have the terminate of odometer to be less than to where it start. Check if the bad input test **Odometer at end** for both the case of less than **Odometer at star** and negative number produce the corresponding error message **(100, -9, 3,john,n,5)**
  3. Have negative number and 0 as the input of rental days to check the boundary produce error message **(0, 100, -9, John, n, 5) and(0, 100, 0, John, n, 5)**
  4. Leave empty string as the input of customer name (0, 100, 3, , n,12) – test if the logic condition in if statement produce the error message
  5. Give a and 1 as the input of **Luxury car? (y/n):**, to test if the logic condition in if statement produce the error message **(0, 100, 9, John, a, 5) and (0, 100, 9, John, 1, 5)**
  6. Give input for month 13 and 0– test the case of out of bound and the boundary conditions**(0, 100, 9, John, 1, 0) and (0, 100, 9, John, 1, 13)**
  7. Input bad data, such that all error messages together to test if only the first error pops up **( -9, -9, -9, , k, 0)**
  8. Input 0 for the mileage total **( 0, 0, 3, John, n, 5)** – test if the base charge is calculated correctly

// season mileage rate test

* 1. Input the month of car driven to be 4 and 11 while other data remain the same. The total cost should be the same to test if the boundary cases from April through November is correct **( 0, 300, 3, John, n, 4) and ( 0, 300, 3, John, n, 11)**
  2. Input number 7 as month **( 0, 300, 7, John, n, 4)** and check if it has the same result as test case i – the test makes sure the domain from 4 to 11 is secure
  3. Input the month of car driven to be 4 and 11 while other data remain the same. The total cost should be the same in order to test if the boundary cases from December through March is correct **( 0, 300, 3, John, y, 12) and ( 0, 300, 3, John, y, 3)**
  4. Input number 1 as month and check if it has the same result as test case k – the test makes sure the domain from 12 to 3 is secure **( 0, 300, 1, John, y, 4)**
  5. Input mileage with 500 miles and in different seasons to check if the seasonal mileage rate is chosen correctly **(0, 500, 3, John, y, 5) and (0, 500, 3, John, y, 1)**
  6. Input mileage with 400 miles and in different seasons to check if the seasonal mileage rate is chosen correctly and if the boundary case is secured **(0, 400, 3, John, y, 5) and (0, 400, 3, John, y, 1)**
  7. Input mileage with 100 miles and in different seasons to check if the seasonal mileage rate is chosen correctly( the seasonal mileage rate should not affect the result) and if the boundary case is secured **(0, 100, 3, John, y, 5) and (0, 100, 3, John, y, 1)**