Assignment 2

Car Rental

***Due:** Tuesday, October 29*

For this assignment we will be writing a Java program to assist with determining whether a customer is eligible for a car rental, and how much money a rental will cost.

Program Summary:

Write a Java program inside a class *CarRental* which asks the customer to answer the following questions:

- How old are you?
- What country will you be renting in?
- Have you ever cause a traffic accident?
- What kind of car would you like to rent?
- How long will you be renting the car for?
- How will you be paying for your car?

After asking all of these questions, you will assess whether or not the customer is eligible to rent a car. If they are eligible, you will compute the cost to rent the car, at which point you will then process their payment, charging them a deposit if they are paying by debit.

Requirements:

The above program must be modularized using the following methods:

char toLowerCase (char)

For many questions, the user will be entering their answers as single chars. In order to make our program case insensitive (so that it works the same regardless of capitalization), toLowerCase will convert any uppercase letters into lowercase letters. If a character that is not a letter is given as input, toLowerCase should return that character as is.

boolean assessmentOfEligibility (int, char, char)

assessmentOfEligibility will return true when the customer is eligible to rent a car, and false otherwise. It takes as input 3 pieces of information, first an int representing the customer's age. Second, a char representing the country where the rental is being made ('c' for Canada, 'u' for United States, 'f' for France). The third input is a char representing if the customer has caused a traffic accident, 'y' for Yes, 'n' for No.

A customer is eligible to rent a car under the following conditions:

If the customer is in Canada, they must be over 21.

If the customer is in the United States, they must be over 18.

If the customer is in France, they must be over 25.

If the customer has been in an accident, they are not eligible to rent a car if they are renting from either Canada or the United States.

double calculateCost (char, int, int)

calculateCost will compute and return the cost to rent the car. It will take as input 3 pieces of information. First, a char representing the type of car being rented ('r' for regular, 'f' for fancy, 'l' for luxurious). The second input is an int representing the number of days which the car will be rented for, the last input is a int representing the customer's age.

To compute the cost of the car, we start with a base price, determined by the type of the car: $40 for regular car, $60 for a fancy car, $80 for a luxurious car. We also add a $20 "underage fee" to the base price of the car if the customer is under 25.

We then determine the rate based off the number of days the car will be rented for. If it is being rented for less than 3 days, we charge 50% of the base price of the car per day. If it is being rented for between 3 and 7 days, we charge 45% of the base price of the car per day. If it is being rented for more than 7 days, we charge 40% of the base price of the car per day.

Example: Ted, who is 23, is renting a fancy car for 6 days. He owes $60 for the base price of the car, plus $20 because he is underage. For each day, he owe 45% of the base price of the car because of the length of his rental, $27 per day times 6 days for a total of $162. The final price of the car is $162 + $60 + $20, for $242.

double calculateDeposit(double, char)

calculateDeposit will calculate and return the deposit owed based upon the cost of the rental and the payment type. It will take as input the calculated cost of the rental for the car, as well as a char representing the type of payment being used ('d' for debit, 'c' for credit).

If they are paying by credit, no deposit is necessary – the method should return 0.

If they are paying by debit, the deposit is either 20% of the cost of the rental, or $100, whichever is greater.

void main(String[])

Your program should include a main method which will accept user input, and call all of these other methods in order to compute the amount owed for the car rental.

The first thing your main method should do is to ask the user the following questions:

- How old are you?
  - (Represented as a integer)
- What country will you be renting in?
  - (Represented as a char – 'c' for Canada, 'u' for United States, 'f' for France)
- Have you ever cause a traffic accident?
  - (Represented as a char – 'y' for yes, 'n' for no)

When you accept user input, you must convert uppercase character to lowercase to make our program case-insensitive. You can assume that your customers are entering valid inputs for all questions. (IE: The country code will always be c, u, or f, the age will always be positive, etc.)

Your method should then check whether the customer is eligible to rent a car using the methods defined above. If they are not eligible, you should immediately print a message saying “Ineligible for car rental”, and your program should terminate.

If they are eligible you should then ask the following questions:

- What kind of car would you like to rent?
  - (Represented as a char – ‘r’ for regular, ‘f’ for fancy, ‘l’ for luxurious)
- How many days will you be renting the car for?
  - (Represented as an integer)

Using this information, you should compute the amount that the rental will cost using the above methods. You should print this amount out to the user, and then ask them:

- How will you be paying for the rental?
  - (Represented as a char – ‘c’ for credit, ‘d’ for debit)

Finally, based on the payment type, you should compute the deposit. You should then print this amount out to the user, along with the total cost (computed as the sum of the rental cost and the deposit) and then congratulate them on their successful car rental.

Before you begin programming:

It is required that you design a flowchart describing each of the methods listed here. You should also write test inputs along with expected results which you can use to test your algorithm to confirm whether or not it gives the correct result.

Summary of what to submit:

You will submit both an electronic and a hard copy piece. The hard copy should have:

a. A table with test input along with the expected results.
b. A flow chart illustrating your algorithm. (Each method should have its ***own flow chart!!!***)
c. A hardcopy print out of the source code.
d. A hardcopy print out of a sample execution.

The electronic copy must include only the CarRental.java file.

Both your program and algorithm must use meaningful variable names in all cases. Your code must also be well commented, and properly indented.