Assignment 3: Credit Card Tester

Assignment for COSC 1200 – Object Oriented Programming I

Prior to attempting this problem, you should have done the following:
1. Viewed the content, including assigned readings for weeks 1 through 8.
2. Viewed the lectures and practiced with the demo code for weeks 1 through 8

General Requirements

1. This assignment can be completed **EITHER** in a group of 2 or individually (your choice).
2. Analyze the problem, it is recommended (but not required, will not be graded) to develop a plan your solution using a flowchart or pseudo code.
3. Submit your solution to the appropriate assignment folder on DC Connect **by the due date provided**.
4. Your instructor will assign a grade and post feedback on your submission to DC Connect.
5. Note: Each class needs comments at the top with at minimum your name(s), the date (date created or date due), a description of what the class does. In addition, it is expected that you provide a sufficient amount of commenting in your code.

Requirements

Complete and submit the program source code that satisfies the following requirements:

For this assignment you are going to create two java classes/files, one named CreditCard.java and another named CardTester.java

1. The CreditCard.java will be used to create objects of this type, this class should have the following components (**Note: this class will not have a main() method**) :
    a. An attribute of type long to store the credit card number, named accountNumber (access type should be private)
    b. An attribute of type int to store the credit card's expiration month, named month (access type should be private)
    c. An attribute of type int to store the credit card's expiration year, named year (access type should be private)
    d. An attribute of type String to store the credit card holder's name, named holder (access type should be private)
    e. A constructor that accepts four arguments, of the types listed above, that places each argument into it's appropriate attribute
    f. A default constructor (i.e. one that does not take any arguments), that will set the attributes to the private class values listed below.
    g. A class variable named DEFAULT_NAME of type String, that contains the value "Bob Loblaws" (access private)

h. A class variable named DEFAULT_ACCOUNT_NUMBER of type long, that contains the value 1234567890123456L (access private)

i. A class variable named DEFAULT_MONTH of type int, that contains the value 1 (zero) (access private)

j. A class variable named DEFAULT_YEAR of type int, that contains the value 99 (access private)

k. One additional class attribute named ACCOUNT_NUMBER_LENGTH that is of type long and that contains the value 16, this number should have public access

l. Create a instance method (i.e. non-static) named displayCardInfo() that will display a specific credit card's information in the form (it does not have to return anything):

> Card Holder Name
>
> Account Number
>
> Month/year

(Note: use a decimal format object to ensure that single digit month and years display a leading zero). i.e. January – month 1 – is displayed "01"

m. A last method named verifyAccountNumber() that returns a boolean (primitive), that checks, using ACCOUNT_NUMBER_LENGTH, whether the number in an objects accountNumber attribute is valid.
   Hint: for this method you will have to use a Wrapper class. Specifically that converts the long stored into a String, and then you can compare the String's length to the class variable's length

2. Create a Java class named CreditCardTester, that has a main method that performs the following:

a. Instantiates three (3) objects of type CreditCard. These objects should be passed the following arguments:

   i. The first one, is not passed arguments, which means they should use the default class values described in part 1g-j

   ii. The second one should be passed the values:
      - "Jane Smith", 5312264554231345, 1, 10 (Jan 2024), note these values are valid

   iii. The third one should be passed the values:
      - "Neva Read", 10234, 5, 8 (April 2022), note the account number is invalid

      b. Each object's information should then be display using the displayCardInfo() method (have a System.out.println before each display to explain which object you are about to display)

      c. Three (3) System.out.println() that tell whether each of the account numbers is valid or not. Only one will be invalid (Jane Read's)

Note: please only submit your .java files to DC Connect.

Style Guide Notes

To be eligible for full marks on this or any assignment in this course your application must conform to the requirements as outlined above as well as the Google style guide as well as appropriate and complete program documentation.