Luhn's Test of Credit Card Numbers (Due 08 Feb 2013)

In this program you will test whether a given 16-digit credit card number is valid or not. Here are the steps in your program:

- You will prompt the user to enter a 16-digit credit card number.
- You will read the credit card number in as a *long* using a Scanner object.
- You will first check if it is a 16-digit number.
- If it is not a 16-digit number your will wirte an error message and exit the program using a *return* statement.
- If it is a 16-digit number you will perform the Luhn's test (outlined below).
- You will then print if the credit card number is valid or not.

You will be making the following assumptions:

- The user enters only digits.
- The user enters digits that can be stored in a *long*.

Here are possible scenarios that may happen. **Your output statements must be exactly the ones specified. Points will be deducted for variations.**

Scenario 1

```
Enter 16-digit credit card number: 123456789876

Not a 16-digit number
```

Scenario 2

```
Enter 16-digit credit card number: 12345678123456780

Not a 16-digit number
```

Scenario 3

```
Enter 16-digit credit card number: 1234567891234567

Invalid credit card number
```

Scenario 4

```
Enter 16-digit credit card number: 4222222222222220

Valid credit card number
```

Luhn's Test: Let us say that the credit card number was made of the following digits:

d1 d2 d3 d4 d5 d6 d7 d8 d9 d10 d11 d12 d13 d14 d15 d16

Then the Luhn's algorithm goes as follows:

- Multiply all the odd digits **d1**, **d3**, … **d15** by 2.
- Sum the digits of each product.
- Now add all the even digits **d2**, … **d16** and the single digit products of the odd digits.
- If the final sum is divisible by 10 then the credit card is valid, otherwise it is invalid.

Here are two examples worked out - one is a valid credit card number and the other is an invalid credit card number.

Valid Credit Number

d1	d2	d3	d4	d5	d6	d7	d8	d9	d10	d11	d12	d13	d14	d15	d16
4	4	3	2	3	3	3	8	8	4	1	3	8	3	4	3
8		6		6		6		16		2		16		8	
8		6		6		6		7		2		7		8	
8	4	6	2	6	3	6	8	7	4	2	3	7	3	8	3

The sum of the digits is **80**. It is divisible by 10 and hence the credit card number is valid.

Invalid Credit Number

d1	d2	d3	d4	d5	d6	d7	d8	d9	d10	d11	d12	d13	d14	d15	d16
8	2	7	3	1	2	3	2	7	3	5	1	0	5	6	9
16		14		2		6		14		10		0		12	
7		5		2		6		5		1		0		3	
7	2	5	3	2	2	6	2	5	3	1	1	0	5	3	9

The sum of the digits is **56**. It is not divisible by 10 and hence the credit card number is invalid.

To obtain the last digit of a number use the modulo (%) operator. To remove the last digit of a number divide by 10. Here is a piece of code that shows how to obtain the last two digits of the credit card number.

```
long creditCard = sc.nextLong();

int d16 = (int) (creditCard % 10);

creditCard = creditCard / 10;

int d15 = (int) (creditCard % 10);
```

The class that you will be writing will be called CreditCard. We will be looking at good documentation, and adherence to the coding convention mentioned below. Your file CreditCard.java will have the following header:

```
/*
  File: CreditCard.java
```

```
  Description:

  Student Name:

  Student UT EID:

  Course Name: CS 312

  Unique Number:

  Date Created:

  Date Last Modified:

*/
```

You will follow the standard Java Coding Conventions. You can either view the HTML page or download the PDF or Postscript and print it out. There is a modification that I would like to make to the standard coding conventions. Please align the opening and closing braces vertically so that you can easily make out the blocks of code. For example:

```
Do this:
if ( x > 5 )
{
  a = b + c;
}

Not this:
if ( x > 5 ) {
  a = b + c;
}
```

Use the turnin program to submit your **.java** file. We should receive your work by 11 PM on Friday, 08 Feb 2013. There will be substantial penalties if you do not adhere to the guidelines.

- You must submit the .java file and not the .class file.
- Your .java file should have the header with the proper documentation.
- You should be submitting your .java file through the web based *turnin* program. We will not accept files e-mailed to us.
- Compile and run your code on the command line.
- Your code must compile before submission.
- Here is the Grading Criteria.