**Institute of Technology Tralee**

**Ord/Hons BSc. in Computing with Specialism (Groups 2+5) - Year 1**

**Continuous Assessment #2**

**Date: 25/3/14**

**Time: 9 – 11 a.m.**

**Object Oriented Programming 1**

**Instructions:** Attempt the following question. You should use the JCreator IDE. When you are finished coding, print out your code in **landscape** format.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Q1.**

You must write a Java program that validates a Visa debit card number that has been issued by the AIB bank in Tralee. In order to be valid, the debit card number must

* Be 16 characters long
* Begin with the digit 4
* Contain all digits
* Satisfy the “golden rule”

In order to demonstrate the “golden rule”, take the following sample debit card number:

4417123456789113

1. The first step is to double every second number beginning with the first number, so we get (numbers spaced out for clarity, with doubled values in bold):

**8** 4 **2** 7 **2** 2 **6** 4 **10** 6 **14** 8 **18** 1 **2** 3

In order to convert the characters in the debit card from a character to its numeric equivalent, you can use the **Character.getNumericValue()**method. You simply pass in the character you have extracted as an argument and it will return the numeric equivalent e.g.

Character.getNumericValue(‘4’) would return the integer value 4, which you can then double.

1. Now simply add **all the digits** of all the numbers above (note that 10 becomes 1+0 = 1 and 14 becomes 1+4 = 5 – hint: you can use the remainder operator **%** to help convert these 2 digit numbers to 1 digit numbers quickly)

The total of all these digits is 70

1. Check to see if the total of all the digits is divisible by 10. If it is then the “golden rule” has been satisfied and the debit card number must be valid.

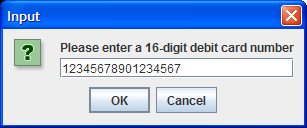
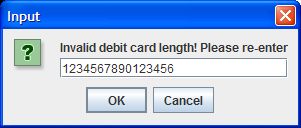
If the debit card number is found to be invalid, your program should issue a suitable error message to the user indicating what was wrong with it and ask the user to re-enter.

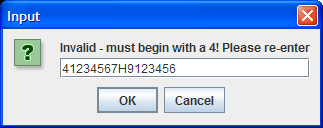
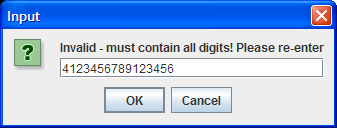
For full marks here your program should, along with a logically correct solution for the problem above, include the usual **single-line** and **multi-line comment** at the top of the program. The multi-line comment should briefly explain the purpose of the program.

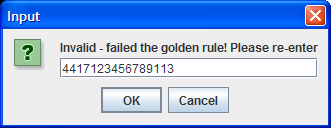
Your program should run as indicated in the following sample screenshots. Also, you should use the test values indicated when testing your own program.

**Sample Screenshots**

**The user gets asked to enter a debit card number. Several invalid ones are entered and the user gets asked to re-enter each time, being told what was wrong with the supplied value:**



**… eventually a valid debit card number is entered and the user gets a confirmation message dialog**

