**Institute of Technology Tralee**

**Ord/Hons BSc. in Computing with Specialism (Group A) - Year 1**

**Continuous Assessment #2**

**Date: 4/12/12**

**Time: 10 – 12 p.m.**

**Introduction to Programming**

**Instructions:** Attempt the following question. You should use the JCreator IDE for coding. When you are finished you must print out your code for correction.

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

**Q1. (Contains modifications to original question)**

Every book published is uniquely identifiable by its ISBN, which is a 10 character string. The first 9 characters of the ISBN are always digits between 0 and 9 and the last character, called the “check digit”, can be either a digit or the character “X” (which is a representation for the numeric value 10).

In order for an ISBN to be valid, apart from the conditions above, it must also abide by the following crucial “golden rule”. The first 9 digits of the ISBN, when multiplied respectively by each of the numbers 10, 9, 8, 7, 6 ….. down to 2 and then added together will produce some total. When this total is divided by 11, then a remainder is produced and subtracting this remainder from 11 must be equal to the check digit.

So, for example, the ISBN 0140124993 is a valid ISBN because it contains 10 characters, all of which are digits and

(0 x 10) + (1 x 9) + (4 x 8) + (0 x 7) + (1 x 6) + (2 x 5) + (4 x 4) + (9 x 3) + (9 x 2) = 118 and

118/11 = 10 remainder 8 and finally

11-8 = 3 which is equal to the check digit.

You must write a Java program that reads in an ISBN value. When the value is entered, it should be firstly validated to ensure that it contains exactly 10 characters. If it does then it should be tested to ensure that the first 9 characters are digits. If it passes this test then it should be tested to ensure the last character is either a digit or an ‘X’. If it fails on any of these tests along the way, the user should receive an appropriate error message. If it passes all of these tests, the ISBN should then be further validated to ensure it abides by the “golden rule”.

Note when doing the multiplication for the “golden rule” test that the characters will need to be converted to their integer equivalents. To do this you must typecast the individual characters as integers and then subtract 48 from the result in each case e.g. to convert the character ‘6’ stored in a char variable ch to the integer value 6, the code would be **(int)ch - 48**

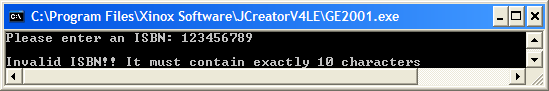
Based on the outcome of the “golden rule” test, the user should receive an appropriate message indicating whether or not the ISBN was valid.

Using the test values as indicated in the screen shots below, the program should give you **exactly** the following output when it runs, including any banners, blank lines, tabs etc.

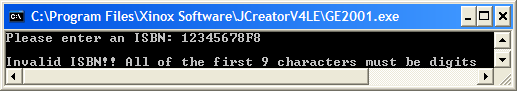
Also note that there will be a few marks awarded for having a **single-line comment** and a **meaningful multi-line comment at the top of the program**

**Sample Screen Shots**

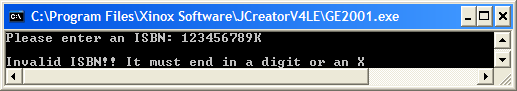
**In this run the user enters a value containing only 9 characters**



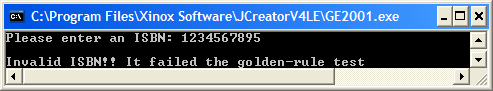
**In this run the value contains 10 characters but not all of the first 9 are digits**



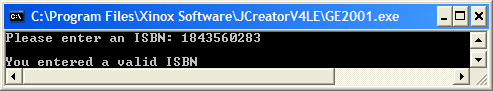
**In this run the value contains 10 characters, the first 9 are all digits but the last one is neither a digit nor an X**



**In this run the value contains 10 characters, all are digits but it fails the golden-rule test**



**In this run the value contains 10 characters, all are digits and it passes the golden-rule test**



**In this run the value contains 10 characters, the first 9 are digits, the last is an X and it passes the golden-rule test**

