**Institute of Technology Tralee**

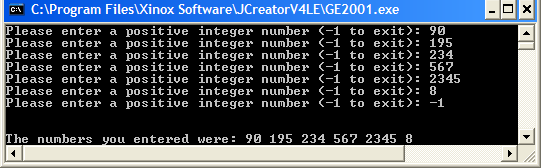
**Computing Department**

**Object Oriented Programming 1**

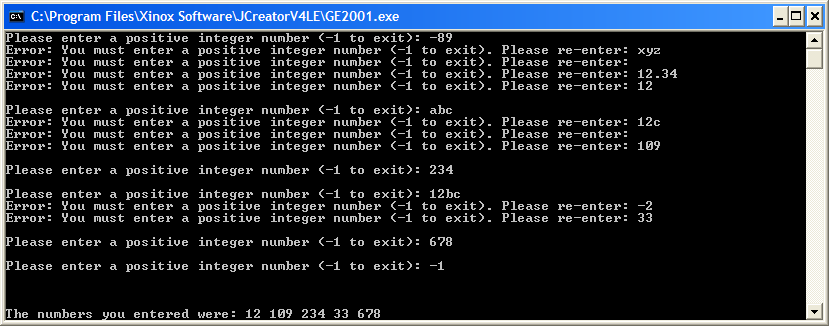
**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Tutorial 6 – Java Loops and Input Validation**

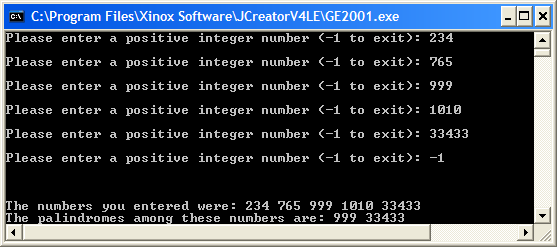
(a) Write a Java program that reads in an arbitrary number of positive integer values, terminated by -1, using a **do-while** loop. When the loop finishes, the program should simply display back on the screen all of the numbers that were entered. Your program should run as indicated in the following sample screenshot:



(b) The program described in part (a) above has no validation code whatsoever. If the user enters negative integer values (other than -1) the loop will continue to function without any difficulty. If floating-point numbers or words are entered, the program will crash at runtime, assuming an integer variable has been used to store the user’s input. Fix this problem now by modifying the program in (a) and writing the necessary validation code to ensure that only positive integer values are accepted by the program. You can assume that any whole numbers entered are within the allowable limits for Java integers, so you don’t need to validate against this possibility.



(c) Add to the program you have developed in (b) above to determine which of the valid numbers entered are *palindromes* i.e. they spell the same backwards and forwards e.g the numbers 12321 and 88 would be a valid palindromes but the numbers 67 and 123 would not be. The program should display these palindromic numbers.



(d) Write a user-defined method that will perform the integer validation referred to in part (b) above. This method should take a String argument which is the String representation of the value entered by the user and return a boolean indicating whether or not the String under consideration is indeed a valid positive integer number. Call the method **validInteger()** and call it within the program as necessary. The program will run exactly as before at runtime as indicated in the following screenshot:

