**Institute of Technology Tralee**

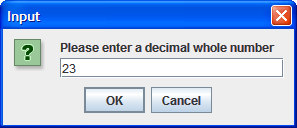
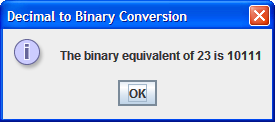
**Computing Department**

**Object Oriented Programming 1**

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

**Tutorial 4 – Methods – Practice for next weeks CA**

**(a)** Write a Java program that makes use of a user-defined method called **decimalToBinary**() that takes a single integer argument representing some positive whole (decimal) number. The method should return the binary equivalent of the value it receives (as a **String**). The main() method should ask the user to enter a decimal whole number (no validation necessary here) and then call decimalToBinary () to determine and return its binary equivalent. The result should be displayed in main() as indicated in the following screenshots:

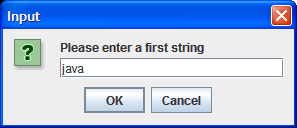
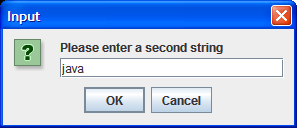
**Hint**: Recall from your computer architecture module that the conversion from decimal to binary involves repeated division of the decimal whole number by 2 and **keep looking at the remainder upon each division** e.g. 23 divided by 2 will give 11 **remainder 1** – 1 now becomes the LSB (least significant bit) in the binary equivalent. Next take 11 and divide by 2, the result is 5 **remainder 1** – so 1 is the next most significant bit in the binary equivalent. Repeated application of this technique until the result upon division becomes zero means that you end up with a series of 1’s and 0’s representing the binary equivalent from MSB to LSB. Keep prepending these bits to a string until complete and then return from the method.

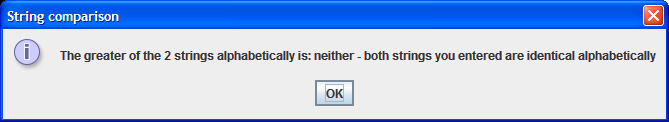
**(b)** Write a Java program that utilizes a user-defined method called **greaterString**() which takes 2 String arguments, supplied by the user. The method should compare the 2 strings and return the string which is greater alphabetically so e.g. the string “orange” would come after the string “apple” alphabetically because the ASCII code of the ‘o’ is greater than the ASCII code of the ‘a’. The main() method here will simply ask the user to enter 2 string values and then call greaterString() to determine and return which of the 2 strings is greater alphabetically. If the 2 strings are identical then neither is greater alphabetically and the method should return this outcome instead.

**Hint**: Have a good think first of all about how you might solve this problem. You’ll want to compare characters in the 2 strings one by one using a looping structure, but break out of the loop as soon as the greater string has been determined. Remember also here that you must be careful when using charAt() as an attempt to extract a character beyond the bounds of the string will result in a **runtime exception** and the program crashes. This would be important in a case where you have 2 strings that begin exactly the same for the entire length of the shorter string e.g. “program” and “programmable”. Try to figure out what you need to do in such cases.

The program should run as indicated in the following sample runs:

**Run1 – here both strings entered are identical**



**Run 2: here the first string (“zebra”) is greater than the second string (“monkey”)**

