**Institute of Technology Tralee**

**Computing Department**

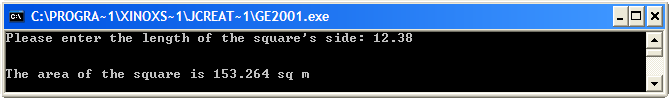
**Object Oriented Programming 1**

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

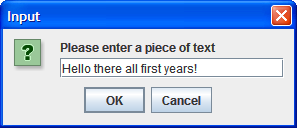
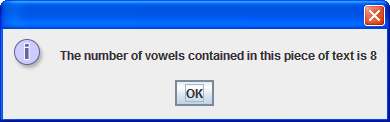
**Tutorial 3 – Methods in Java**

**(a)** Write a Java program that utilises a user-defined Java method called **squareArea**() that takes a single argument representing the length of the side of the square (this value could be fractional). The method should just return the area of the square. The main() method is to be written so that it asks the user to supply the length of the squares side and then calls squareArea() to determine and return its area. The main() will display the area to **3 decimal places** and the program will run as indicated in the sample screenhot below:

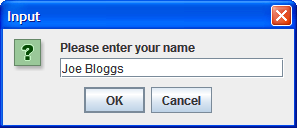
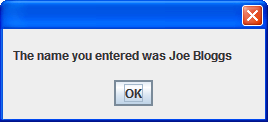
N.B. Area of square = (length of side)2



**(b)** Write a Java program that utilises a user-defined method called **vowelCount**() that takes a single String argument representing a user-supplied text value. The method should return the number of vowels contained in the piece of text supplied. The main() method is to be written so that it asks the user to supply a piece of text and then calls vowelCount() to determine and return the number of vowels it contains. The number of vowels then gets displayed by main(), as indicated in the sample screenshots below:

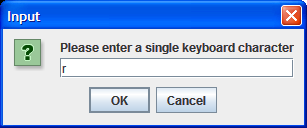
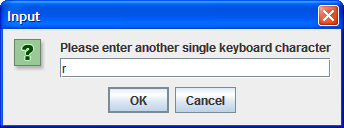
 

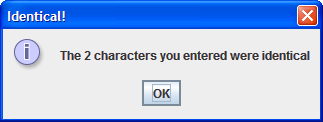
**(c)** Write a Java program that makes use of a user-defined method called **getUserName**() that takes no arguments. The method should simply ask the user to enter their name in an input dialog and return that name. The main() method should call getUserName() to determine the user’s name and then display it in a message dialog as indicated in the following screenshots:

**(d)** Write a Java program that makes use of a user-defined method called **sameCharacter**() that takes 2 char arguments and determines whether the two arguments represent the same character. The method will return the boolean value true or false to indicate the outcome. In the main() method, the user should be asked to enter (separately) 2 single keyboard characters. Once entered, the main() should then call sameCharacter() to determine whether or not the characters entered were identical and then display the result, as indicated in the following sample screenshots:

**Run 1 – both characters the same**



**Run 2 –characters different**

