**Institute of Technology Tralee**

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

**Continuous Assessment #3**

**Date: 28/4/14**

**Time: 2 – 4 p.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.**

Write a Java program that contains a JFrame window. This window should contain 2 labels, a text-field (of size 10) and a button. It should have dimensions of 300 x 100 pixels. It should use a flow-layout style and the application should terminate when the close button is hit on its title bar. The first label should originally have the text “No captcha yet generated” on it when the application is first launched. When the button gets pressed, a 10 character “captcha” should get randomly generated and displayed on the first label.

The “captcha” should only contain letters (uppercase or lowercase) or digits.

In order to generate the “captcha” string, you should use the fact that every character has a unique ASCII code number. The ASCII code values that you are interested in here are:

* 48-57 for the characters ‘0’ to ‘9’
* 65-90 for the characters ‘A’ to ‘Z’
* 97-122 for the characters ‘a’ to ‘z’

So you want to use a loop that will generate randomly one of the numbers above each time it iterates and then build a string out of the character equivalents by type-casting each of the numbers to their char equivalent e.g. imagine that the random number 104 gets generated – we convert that to its char equivalent by using **(char) 104** which gives us lowercase ‘h’. We then add that on to an existing string to build up the “captcha” string as we go along.

The second label should merely act as a prompt to the user to ask them to enter the “captcha” that got generated into the associated text-field.

When the user hits return on the text-field, a message dialog will get displayed indicating whether or not the value entered by the user matched exactly the generated “captcha”.

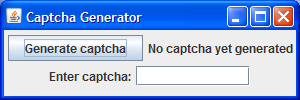
Note that there is **no validation** whatsoever required for this CA.

You should make use of the **appendix** of method definitions from the X: drive when answering this question.

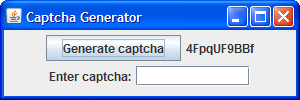
Some sample runs of the program are as illustrated below.

**Sample Screenshots**

**After launching the application it appears as follows**



**Each time the user presses the button, a randomly generated “captcha” gets created and this string is displayed on the first (top) label**



**If the user enters something (or enters nothing) into the text-field and hits return on it, the value entered by the user gets compared to the generated “captcha” and a message dialog indicates whether they match or not**

