**Institute of Technology Tralee**

**Computing Department**

**Object Oriented Programming 1**

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

**Session 7 – Review of some of last year’s CAs on**

**GUI**

Write a Java program that contains a JFrame window. This window should contain 1 label and 2 buttons. It should have dimensions of 230 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 “Your starting balance is €10” on it when the application is first launched.

This GUI aims to play a simple game where a random whole number between 1 and 100 inclusive gets generated each time the user presses either the “Odd” or “Even” button, to indicate whether they think the number generated will be odd or even. If the user guesses correctly, their balance increases by €1, if they are wrong it gets reduced by €1 instead.

The user can keep playing the game until their balance becomes zero, at which time a message dialog will appear giving the user an appropriate error message and, when the user presses OK on this dialog, the application will terminate immediately.

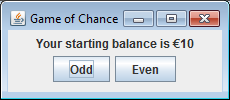
Note that there is **no validation** whatsoever required for this CA, nor are any looping structures required.

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

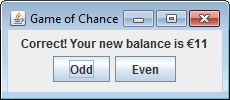
Some sample runs of the program are as illustrated below.

**Sample Screenshots**

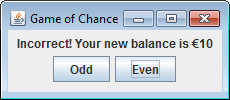
**After launching the application it appears as follows**



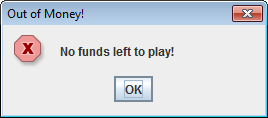
**Each time the user presses the “Odd” or “Even” button, a random number between 1 and 100 gets generated and then it is tested to see if it is odd or even. If it matches the user’s guess then the balance will increase by €1 and the label text will change to reflect this as follows:**



**If the user’s guess is wrong, then the balance will decrease by €1 and the label text will change to reflect this as follows:**



**If the user has a bad run of luck then their balance will eventually reach zero, at which point the following message dialog appears and the game terminates as soon as the user presses OK on the dialog (hint: set the initial balance to 1 temporarily in order to test out this functionality, otherwise the balance may never reach zero)**



**(b)**

Write a Java program that contains a JFrame window. This window should contain a label, a text-field (of size 20) and a text-area. It should have dimensions of 400 x 150 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 just acts as a prompt and should have the words “Enter Text” on it. When the application is first launched, the text area originally has the words “No text entered yet” on it.

The user will enter some text into the text-field and, when the user then hits return on the text-field, the text entered will be analysed to determine

* The total number of characters in the text
* The number of lowercase letters in the text
* The number of uppercase vowels in the text

The results of this analysis will then get displayed on the text-area. The user can keep interacting with the GUI after this, analysing new pieces of text. The text-area here will use a “monospaced” font, the style will be plain font and the point size for the font will be 12 in this case. No special formatting needs to be used on the text-area otherwise for this CA.

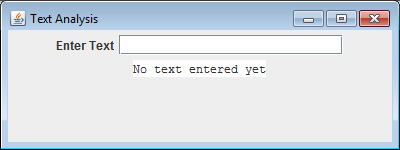
Note that there is **no validation** whatsoever required for this CA, but that a loop is needed within the event-handling code in order to complete some of the text analysis.

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

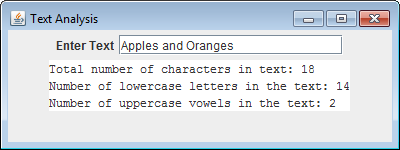
Some sample runs of the program are as illustrated below.

**Sample Screenshots**

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



**When the user enters some text and hits return on the text-field, the text gets analysed and the results of the analysis get displayed on the text-area as follows:**



**The user can just keep interacting with the GUI, entering more text and getting it analysed**

