ENGINEERING 1D04, Lab 11: Objects

This assignment has to be submitted via ELM before the end of the lab session. If you are not done the lab, please finish it later and submit it using ELM. Although the submission for this lab is not graded, it is important that you practice the submission process so that you are ready for next week's marked lab. The process of submission is slightly different from previous weeks because you will be starting with template code.

Assignment: Number Game.

A zip file containing a template for the solution of this assignment is available in the ELM assignment section. This includes the form shown in Figure 1. The assignment involves a simple number guessing game.

Figure 1

Brief description of game: User specifies a minimum and maximum range for a number. The computer generates a random number, r, in the range minimum $\leq$ r $\leq$ maximum. The user then guesses the "secret" number. If the user guesses correctly, a label under the guess text box must show "You guessed correctly." If not, the label shall show "Guess Higher!" or "Guess Lower!" as appropriate. The form is set up to implement two games simultaneously. The user can play either game at any time. Both games can be in progress at the same time.

The two games display slightly different information – but the games operate the same way. In game 1, the user makes a guess and cannot see how many guesses are left until there is only 1 guess left. When there is only 1 guess left, a label displaying "This is your last guess" is displayed. In game 2, the user makes a guess and the number of guesses left is automatically shown in the relevant label. In this game, if the game is over and the number has not been guessed correctly, a message box showing the secret number shall be displayed.

To save time, you do not need to protect against invalid input in the text boxes.

The solution template sets up a class that MUST be used to implement the above form. The class method stubs are included and comments above the method specify the behaviour that must be implemented in that method. This will provide enough detail to determine precisely how the games must operate. In other words, complete the class first and then use the methods in the class to calculate any values you need on the form.

This assignment is not to be removed from the lab

Information on using the random number class:

Visual C# includes a class for random number generation. The class is called *Random*.
We can create random number objects by `Random r = new Random();`

Assuming we want to work with integer random numbers, we can then generate random numbers by successive calls to `r.Next(m)`, where m is an integer. This generates a random number in the range [0, m-1]. So, if q is an integer, `q = r.Next(6);` generates a value of q such that $0 \leq q \leq 5$.

Submission details:

Important: Click "Save All" to save your project before compressing, uploading and submitting it

Implementation in Visual C#. You may edit the project from the zip file stored on ELM. When you are ready to submit, please compress the folder as usual, but then rename the zip file "MacID_StudentNumber_LabSection_Lab11", where MacID, Student Number and LabSection should be the values that correspond to you. To rename a file you can right click on it and then select the "Rename" option.