Exercise 3b: Numerical input

1. Objectives

The main objectives of this exercise are:

- To create a Windows Desktop application with graphical user interface (GUI)
- Read a number from a text box, validate and do some calculation.

2. To Do

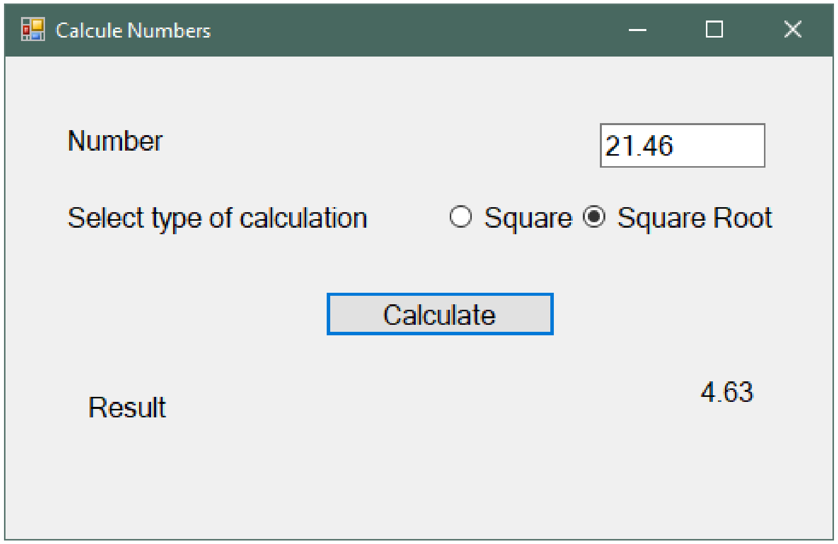
2.1 Create a Windows Forms application and design a simple GUI as shown in the figure below. The application should allow the user to give a numerical value. The value given by the user is actually a series of characters, i.e. a string. The program should convert it to a floating-point number (double), before it can be saved in a variable of double type. Use labels for non-editable texts and a textbox for the input. The output is to be calculated and displayed when the user presses the button.

Design your GUI, run and test the program so it works well.

2.2 The best way to convert a text representation of a numerical value into a number is to use the TryParse method of the floating-point type you are using. In this exercise, we are using a double type and hence we are going to use double.TryParse. This method returns a true value if the conversion can be performed and false otherwise. Using this feature, you can prevent the program from undesired termination. The syntax for this method:

```
bool double.TryParse (string s, out double result);
```

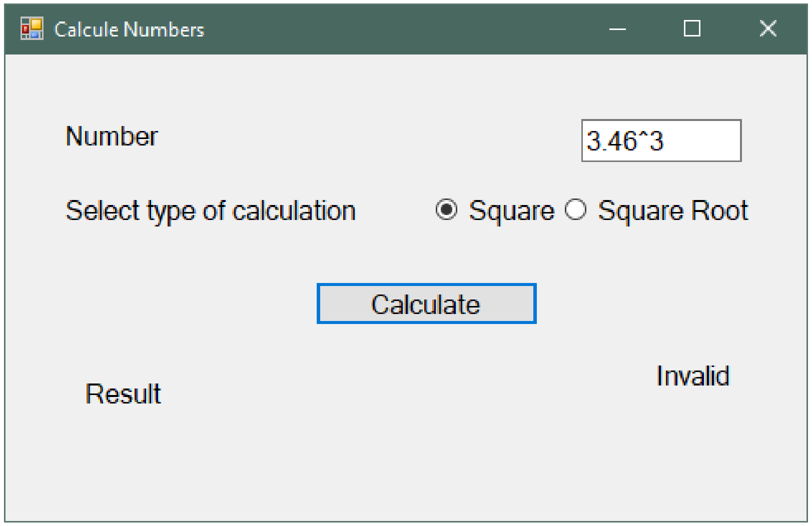
The parameter s can be the contents of a textbox, which must be in a valid numerical format according to C#'s rules. The converted value will be saved in the out variable (result).



Number 21.46

Select type of calculation ○ Square ● Square Root

Result 4.63



Number 3.46^3

Select type of calculation ● Square ○ Square Root

Result Invalid

3. Solution

A soundless video showing how to program this exercise step-by-step is available in the module.

Good Luck!