# Unit 5 - Programming Projects

Create a new project/solution for each of the following projects. Name the solution Unit?Project? and the project Project?.

Example:

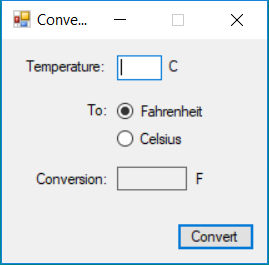
* Solution: Unit1Project4
* Project: Project4

## Windows Applications and Controls

### Project 1

Write a temperature conversion program that converts from Fahrenheit to Celsius and vice versa. The temperature will be entered into a TextBox control. A Label will be used to display the converted temperature. Two RadioButton controls will indicate which temperature type to be converted to. A Button control labeled “Convert” will update the temperature output label.

### Form Design and Initial State



#### **Form Design**

* The title of the form is “Converter”.
* The form cannot be maximized.
* Temperature Textbox has focus.
* Fahrenheit is selected.
* The text in the converted output Label is aligned to the right.
* The “Convert” button is the default button.

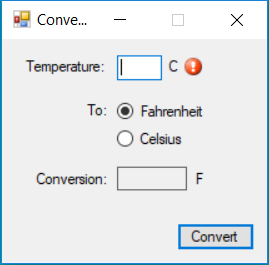
**Note**: Styling of controls may look different on your OS.

#### **Functionality**

When the Button is clicked, validate the temperature input to ensure it is a numeric value.

If the temperature input is not numeric:

* display an error icon beside the temperature TextBox with the message “Temperature cannot contain letters or special characters.”
* update the converted temperature Label, such that it contains no text (to indicate to the user that the conversion was not possible).
* set focus to the temperature Textbox and select all it’s text.



When the a successful conversion is made, ensure all ErrorProvider icons are not visible on the form and update the conversion Label such that it displays the converted temperature.

When text is entered into the TextBox, update the converted temperature Label such that it contains no text. This is a visual reminder to the user that the convert Button must be clicked to perform the conversion.

When either RadioButton is checked:

* Update the temperature symbol Labels (these Labels are located to the right of the temperature TextBox and Output label and contain the text “C” or “F”)
* If a successful conversion has taken place, update the converted temperature Label.

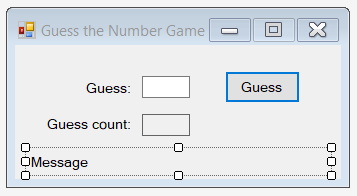
### Project 2

Update Project 1 to use a NumericUpDown control in place of the Textbox. The NumericUpDown will allow the user to select a value between -40 and 40 with an increment of 1. If a temperature has been converted, updating the temperature value will update the converted temperature Label.

### Project 3

Write a “guess the number” program. Your program will choose a random integer between 1-1000. The user will enter their guess into a TextBox control. A Button control will be used to accept the guess and compare it to the secret number. Pressing the Enter key on the keyboard will also accept the guess. Form Design

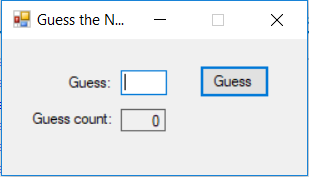
#### **Form Design**



**Note**: This image is from design view in Visual Studio.

On form startup, the form and it’s controls will have the following state:

* Form title: “Guess the Number Game”.
* Form cannot be resized. (BorderStyle property).
* Form’s maximize button disabled.
* Guess Textbox: No text, has focus.
* Guess count output label: Contains the the text “0” and the text is right aligned horizontally and middle vertically.
* Message Label: No text.



**Note**: Styling of controls may look different on your OS. Functionality

When the text is changed in the Textbox, ensure there is no text in the message label.

When the button is clicked validate the input (guess). The guess must be whole number between 1-1000. If the input is invalid:

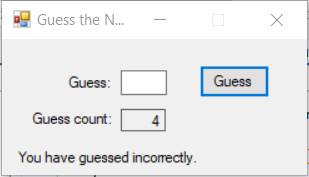
* Display an error icon beside the Textbox with the message “Invalid guess.”.
* Set focus to the Textbox and select its text.

If the input is valid:

* Clear the text in the Textbox.
* Increment the guess count and update the guess count label.
* Determine if the guess is correct.

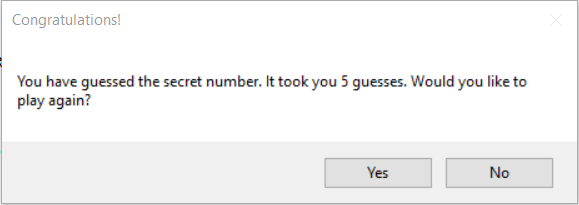
If the user has guessed the secret number incorrectly:

* Update the message label with the text “You have guessed incorrectly.”
* Set focus to the Textbox.



If the user has guessed the secret number correctly:

* Display a dialog box congratulating the user for guessing correctly, indicate how many guesses it took them, and ask them to play again. If they wish to play again, reset the form back to its initial state with a new secret number. If they do not wish to play again, close the form.



### Project 4

Update Project 3 to include a Checkbox which will indicate if the user wishes to play in “Easy Mode”. Easy mode will add a hint to the “incorrect guess” message that is outputted to the message label. Append the following:

* If the guess is greater than the secret number, append “Your guess is too high.”
* Otherwise, append “Your guess is too low.”

Ensure that after the first guess is made, the Checkbox is disabled. If the user plays the game again, enable the Checkbox.