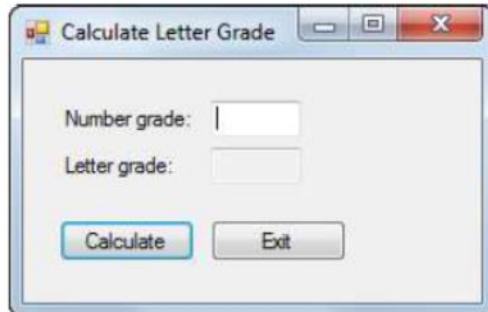


## Extra 2-1 Design a simple form

In this exercise, you'll design a form that lets the user enter a number grade and then displays the letter grade when the user clicks the Calculate button.



1. Start a new project named CalculateLetterGrade in the Extra Exercises\Chapter 02\CalculateLetterGrade directory. Be sure to store the solution in its own directory.
2. Add the labels, text boxes, and buttons to the form as shown above. Then, set the properties of these controls as follows:

Default name	Property	Setting
label1	Text	&Number grade:
	TextAlign	MiddleLeft
	TabIndex	0
label2	Text	Letter grade:
	TextAlign	MiddleLeft
textBox1	Name	txtNumberGrade
	TabIndex	1
textBox2	Name	txtLetterGrade
	ReadOnly	True
	TabStop	False
button1	Name	btnCalculate
	Text	&Calculate
	TabIndex	2
button2	Name	btnExit
	Text	E&xit
	TabIndex	3

3. Now, set the properties of the form as follows:

Default name	Property	Setting
Form1	Text	Calculate Letter Grade
	AcceptButton	btnCalculate
	CancelButton	btnExit
	StartPosition	CenterScreen

4. Use the Form Designer to adjust the size and position of the controls and the size of the form so they look as shown above.
5. Rename the form to frmCalculateGrade. When you're asked if you want to modify any references to the form, click the Yes button.
6. Save the project and all of its files.

## Extra 3-1    Code and test the Calculate Letter Grade form

---

In this exercise, you'll add code to the Calculate Letter Grade form that you designed in extra exercise 2-1. Then, you'll build and test the project to be sure it works correctly.

1. Open the CalculateLetterGrade project in the Extra Exercises\Chapter 03\CalculateLetterGrade directory.
2. Display the form in the Form Designer, and double-click the Calculate button to generate a Click event handler for it. Then, add this statement to the event handler to get the number grade the user enters:

```
decimal numberGrade = Convert.ToDecimal(txtNumberGrade.Text);
```

3. Add this statement to the event handler to declare and initialize the variable that will hold the letter grade:

```
string letterGrade = "";
```

Then, add this if-else statement to set the letter grade:

```
if (numberGrade >= 88)
{
    letterGrade = "A";
}
else if (numberGrade >= 80 && numberGrade <= 87)
{
    letterGrade = "B";
}
else if (numberGrade >= 68 && numberGrade <= 79)
{
    letterGrade = "C";
}
else if (numberGrade >= 60 && numberGrade <= 67)
{
    letterGrade = "D";
}
else
{
    letterGrade = "F";
}
```

4. Add this statement to display the letter grade in the Letter Grade text box:
5. Finally, add this statement to move the focus back to the Number Grade text box:

```
txtLetterGrade.Text = letterGrade;
```

```
txtNumberGrade.Focus();
```

6. Return to the Form Designer, and then double-click the Exit button to generate a Click event handler for it. Then, add this statement to the event handler to close the form:

```
this.Close();
```

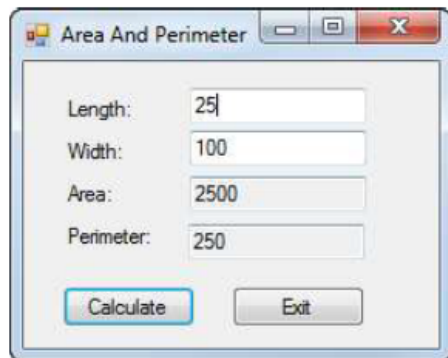
7. Run the application, enter a number between 0 and 100, and then click the Calculate button. A letter grade should be displayed and the focus should return to the Number Grade text box. Next, enter a different number and press the enter

key to display the letter grade for that number. When you're done, press the Esc key to end the application.

## Extra 4-1 Calculate area and perimeter

---

In this exercise, you'll create a form that accepts the length and width of a rectangle from the user and then calculates the area and perimeter of the rectangle.



Length:	25
Width:	100
Area:	2500
Perimeter:	250


Calculate Exit

1. Start a new project named AreaAndPerimeter in the Extra Exercises\Chapter 04\AreaAndPerimeter directory.
2. Add labels, text boxes, and buttons to the default form and set the properties of the form and its controls so they appear as shown above. When the user presses the Enter key, the Click event of the Calculate button should fire. When the user presses the Esc key, the Click event of the Exit button should fire.
3. Create an event handler for the Click event of the Calculate button. This event handler should get the values the user enters for the length and width, calculate and display the area (length x width) and perimeter ( $2 \times \text{length} + 2 \times \text{width}$ ), and move the focus to the Length text box. It should provide for decimal entries, but you can assume that the user will enter valid decimal values.
4. Create an event handler for the Click event of the Exit button that closes the form.
5. Test the application to be sure it works correctly.

## Extra 4-2    Accumulate test score data

---

In this exercise, you'll create a form that accepts one or more scores from the user. Each time a score is added, the score total, score count, and average score are calculated and displayed.

A screenshot of a Windows-style application window titled "Score Calculator". The window has a standard title bar with minimize, maximize, and close buttons. Inside the window, there are four text boxes with labels to their left: "Score:" containing "98", "Score total:" containing "472", "Score count:" containing "5", and "Average:" containing "94". To the right of the "Score:" text box is a button labeled "Add". At the bottom of the window are two buttons: "Clear Scores" on the left and "Exit" on the right.

1. Start a new project named ScoreCalculator in the Extra Exercises\Chapter 04\ScoreCalculator directory.
2. Add labels, text boxes, and buttons to the default form and set the properties of the form and its controls so they appear as shown above. When the user presses the Enter key, the Click event of the Add button should fire. When the user presses the Esc key, the Click event of the Exit button should fire.
3. Declare two class variables to store the score total and the score count.
4. Create an event handler for the Click event of the Add button. This event handler should get the score the user enters, calculate and display the score total, score count, and average score, and move the focus to the Score text box. It should provide for integer entries, but you can assume that the user will enter valid integer values.
5. Create an event handler for the Click event of the Clear Scores button. This event handler should set the two class variables to zero, clear the text boxes on the form, and move the focus to the Score text box.
6. Create an event handler for the Click event of the Exit button that closes the form.
7. Test the application to be sure it works correctly.