Extra 5-1 Calculate the factorial of a number

In this exercise, you’ll create a form that accepts an integer from the user and then calculates the factorial of that integer.

Graphical user interface, application

Description automatically generated

The factorial of an integer is that integer multiplied by every positive integer less than itself. A factorial number is identified by an exclamation point following the number. Here’s how you calculate the factorial of the numbers 1 through 5:

1! = 1 which equals 1  
2! = 2 \* 1 which equals 2  
3! = 3 \* 2 \* 1 which equals 6  
4! = 4 \* 3 \* 2 \* 1 which equals 24  
5! = 5 \* 4 \* 3 \* 2 \* 1 which equals 120

To be able to store the large integer values for the factorial, this application can’t use the Int32 data type.

1. Start a new project named Factorial in the Extra Starts\Chapter 05 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 number the user enters, calculate the factorial of that number, display the factorial with commas but no decimal places, and move the focus to the Number text box. It should return an accurate value for integers from 1 to 20. (The factorial of the number 20 is shown in the form above.)
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 5-2 Calculate change

In this exercise, you’ll develop a form that tells how many quarters, dimes, nickels, and pennies are needed to make change for any amount of change from 0 through 99 cents. One way to get the results is to use the division and modulus operators and to cast the result of each division to an integer.

Graphical user interface, application

Description automatically generated

1. Start a new project named ChangeCalculator in the Extra Starts\Chapter 05 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. Then, write the code for calculating and displaying the number of quarters, dimes, nickels, and pennies that are needed for the change amount the user enters. This code should provide for integer entries, but you can assume that the user will enter valid integer 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 5-3 Calculate income tax

In this exercise, you’ll use nested if statements and arithmetic expressions to calculate the federal income tax that is owed for a taxable income amount entered by the user.

Graphical user interface, application

Description automatically generated

This is the 2020 table for the federal income tax on individuals that you should use for calculating the tax:

|  |  |  |  |
| --- | --- | --- | --- |
| **Taxable income** | | **Income tax** | |
| **Over…** | **But not over…** |  | **Of excess over…** |
| $0 | $9,875 | $0 plus 10% | $0 |
| $9,876 | $40,125 | $987.50 plus 12% | $9,875 |
| $40,126 | $85,525 | $4,617.50 plus 22% | $40,125 |
| $85,526 | $163,300 | $14,605.50 plus 24% | $85,525 |
| $163,301 | $207,350 | $33,271.50 plus 32% | $163,300 |
| $207,351 | $518,400 | $47,367.50 plus 35% | $207,350 |
| $518,401 |  | $156,235 plus 37% | $518,400 |

1. Start a new project named TaxCalculator in the Extra Starts\Chapter 05 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 Exit button that closes the form.
4. Create an event handler for the Click event of the Calculate button. Then, write the code for calculating and displaying the tax owed for any amount within the first two brackets in the table above. This code should provide for decimal entries, but you can assume that the user will enter valid decimal values. To test this code, use income values of 8700 and 35350, which should display taxable amounts of 870 and 4044.50.
5. Add the code for the next tax bracket. Then, if you have the time, add the code for the remaining tax brackets.