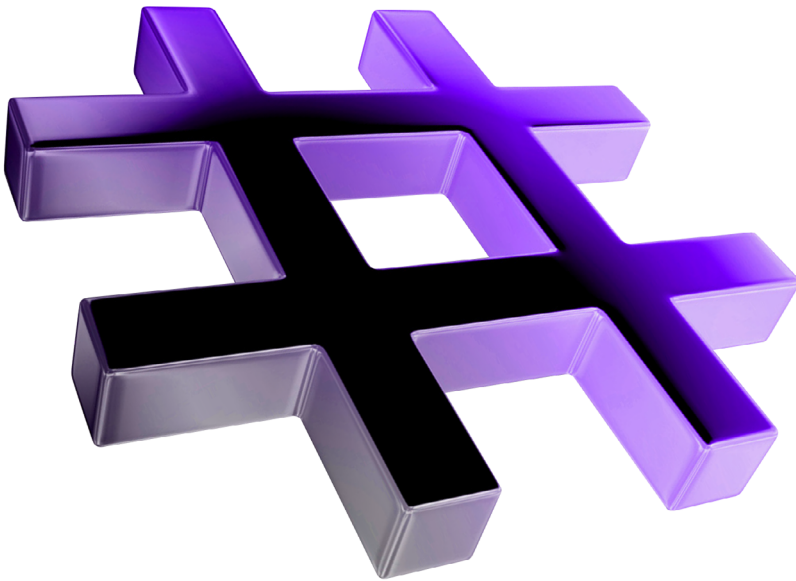# Introduction to C# App Programming

3

Objectives

In this chapter you'll:

- Write simple C# apps using code rather than visual programming.

- Input data from the keyboard and output data to the screen.

- Use C# 6's `string` interpolation to create formatted `string`s by inserting values into `string` literals.

- Declare and use data of various types.

- Store data in memory and retrieve it.

- Use arithmetic operators.

- Determine the order in which operators are applied.

- Write decision-making statements with equality and relational operators.

Self-Review Exercises

3.1   Fill in the blanks in each of the following statements:
  a) A(n) _____ begins the body of every method, and a(n) _____ ends the body of every method.
  ANS: left brace ({), right brace (}).
  b) Most statements end with a(n) _____.
  ANS: semicolon (;).
  c) The _____ statement is used to make decisions.
  ANS: if.
  d) _____ begins a single-line comment.
  ANS: //.
  e) _____, _____ and _____ are called whitespace. Newline characters are also considered whitespace.
  ANS: Blank lines, space characters, tab characters.
  f) _____ are reserved for use by C#.
  ANS: Keywords.
  g) C# apps begin execution at method _____.
  ANS: Main.
  h) Methods _____ and _____ display information in the console window.
  ANS: Console.WriteLine and Console.Write.
  i) _____ enables you to insert values directly into a string literal.
  ANS: string interpolation.

3.2   State whether each of the following is *true* or *false*. If *false*, explain why.
  a) Comments cause the computer to display the text after the // on the screen when the app executes.
  ANS: False. Comments do not cause any action to be performed when the app executes. They're used to document apps and improve their readability.
  b) C# considers the variables number and NuMbEr to be identical.
  ANS: False. C# is case sensitive, so these variables are distinct.
  c) The remainder operator (%) can be used only with integer operands.
  ANS: False. The remainder operator also can be used with noninteger operands in C#.
  d) The arithmetic operators *, /, %, + and – all have the same level of precedence.
  ANS: False. The operators *, / and % are on the same level of precedence, and the operators + and – are on a lower level of precedence.
  e) An interpolated string must begin with a $ before the string literal.
  ANS: True.

3.3   Write statements to accomplish each of the following tasks:
  a) Declare variables c, thisIsAVariable, q76354 and number to be of type int.
  ANS:
```
int c;
int thisIsAVariable;
int q76354;
int number;
```
  b) Prompt the user to enter an integer.
  ANS: Console.Write("Enter an integer: ");
  c) Input an integer and assign the result to int variable value.
  ANS: value = int.Parse(Console.ReadLine());

d) If the variable `number` is not equal to 7, display `"The variable number is not equal to 7"`.

ANS: `if (number != 7)`

```
    {
        Console.WriteLine("The variable number is not equal to 7");
    }
```

e) Display `"This is a C# app"` on one line in the console window.

ANS: `Console.WriteLine("This is a C# app");`

f) Display `"This is a C# app"` on two lines in the console window. The first line should end with `C#`. Use method `Console.WriteLine`.

ANS: `Console.WriteLine("This is a C#\napp");`

g) Write a statement that uses `string` interpolation to display the sum of the variables `x` and `y`. Assume variables `x` and `y` of type `int` exist and already have values.

ANS: `Console.WriteLine($"The sum of {x} and {y} is {x + y}");`

3.4 Identify and correct the errors in each of the following statements:

a) `if (c < 7);`

```
    {
        Console.WriteLine("c is less than 7");
    }
```

ANS: *Error:* Semicolon after the right parenthesis of the condition `(c < 7)` in the `if` statement.
Correction: Remove the semicolon after the right parenthesis. [*Note*: With the semicolon, the output statement executes regardless of whether the condition in the `if` is true.]

b) `if (c => 7)`

```
    {
        Console.WriteLine("c is equal to or greater than 7");
    }
```

ANS: *Error:* The relational operator `=>` is incorrect.
Correction: Change `=>` to `>=`.

3.5 Write declarations, statements or comments that accomplish each of the following tasks:

a) State that an app will calculate the product of three integers.

ANS: `// Calculating the product of three integers`

b) Declare the variables `x`, `y`, `z` and `result` to be of type `int`.

ANS: `int x;`

```
    int y;
    int z;
    int result;
```

c) Prompt the user to enter the first integer.

ANS: `Console.Write("Enter first integer: ");`

d) Read the first integer from the user and store it in the variable `x`.

ANS: `x = int.Parse(Console.ReadLine());`

e) Prompt the user to enter the second integer.

ANS: `Console.Write("Enter second integer: ");`

f) Read the second integer from the user and store it in the variable `y`.

ANS: `y = int.Parse(Console.ReadLine());`

g) Prompt the user to enter the third integer.

ANS: `Console.Write("Enter third integer: ");`

h) Read the third integer from the user and store it in the variable `z`.

ANS: `z = int.Parse(Console.ReadLine());`

      i) Compute the product of the three integers contained in variables x, y and z, and assign the result to the variable result.

      ANS: result = x * y * z;

      j) Display the message "Product is", followed by the value of the variable result—use string interpolation.

      ANS: Console.WriteLine($"Product is {result}");

3.6   Using the statements you wrote in Exercise 3.5, write a complete app that calculates and displays the product of three integers.

      ANS: The solution to Self-Review Exercise 3.6 is as follows:

```
1   // Exercise 3.6: Product.cs
2   // Calculating the product of three integers.
3   using System;
4
5   class Product
6   {
7      static void Main()
8      {
9         int x; // stores first number to be entered by user
10        int y; // stores second number to be entered by user
11        int z; // stores third number to be entered by user
12        int result; // product of numbers
13
14        Console.Write("Enter first integer: "); // prompt for input
15        x = int.Parse(Console.ReadLine()); // read first integer
16
17        Console.Write("Enter second integer: "); // prompt for input
18        y = int.Parse(Console.ReadLine()); // read second integer
19
20        Console.Write("Enter third integer: "); // prompt for input
21        z = int.Parse(Console.ReadLine()); // read third integer
22
23        result = x * y * z; // calculate the product of the numbers
24        Console.WriteLine($"Product is {result}");
25     } // end Main
26  } // end class Product
```

```
Enter first integer: 10
Enter second integer: 20
Enter third integer: 30
Product is 6000
```

Fig. 3.1 | Solution to Exercise 3.6.

Exercises

*Note: Solutions to the code exercises are located in the **sol_ch03** folder*

3.7   Fill in the blanks in each of the following statements:

    a) _____ are used to document an app and improve its readability.

    ANS: Comments.

    b) A decision can be made in a C# app with a(n) _____.

    ANS: if statement.

    c) Calculations are normally performed by _____ statements.

    ANS: assignment.

    d) The arithmetic operators with the same precedence as multiplication are _____ and
        _____.
    ANS: division (/), remainder (%)
    e) When parentheses in an arithmetic expression are nested, the _____ set of parenthe-
        ses is evaluated first.
    ANS: innermost.
    f) A location in the computer's memory that may contain different values at various times
        throughout the execution of an app is called a(n) _____.
    ANS: innermost.

3.8 Write C# statements that accomplish each of the following tasks:
    a) Display the message `"Enter an integer: "`, leaving the cursor on the same line.
    ANS: `Console.Write("Enter an integer: ");`
    b) Assign the product of variables `b` and `c` to variable `a`.
    ANS: `a = b * c;`
    c) State that an app performs a simple payroll calculation (i.e., use text that helps to doc-
        ument an app).
    ANS: `// This application performs a simple payroll calculation.`

3.9 State whether each of the following is *true* or *false*. If *false*, explain why.
    a) C# operators are evaluated from left to right.
    ANS: False. Some operators (e.g., assignment, =) evaluate from right to left.
    b) The following are all valid variable names: `_under_bar_`, `m928134`, `t5`, `j7`, `her_sales`,
        `his_account_total`, `a`, `b`, `c`, `z` and `z2`.
    ANS: True.
    c) A valid C# arithmetic expression with no parentheses is evaluated from left to right.
    ANS: False. The expression is evaluated according to operator precedence and associativity.
    d) The following are all invalid variable names: `3g`, `87`, `67h2`, `h22` and `2h`.
    ANS: False. Identifier `h22` is a valid variable name.

3.10 Assuming that `x = 2` and `y = 3`, what does each of the following statements display?
    a) `Console.WriteLine($"x = {x}");`
    ANS: `x = 2`
    b) `Console.WriteLine($"Value of {x} + {x} is {x + x}");`
    ANS: `Value of 2 + 2 is 4`
    c) `Console.Write("x =");`
    ANS: `x =`
    d) `Console.WriteLine($"{x + y} = {y + x}");`
    ANS: `5 = 5`

3.11 Which of the following C# statements contain variables whose values are modified?
    a) `p = i + j + k + 7;`
    b) `Console.WriteLine("variables whose values are modified");`
    c) `Console.WriteLine("a = 5");`
    d) `value = int.Parse(Console.ReadLine());`
    ANS: (a) and (d)

3.12 Given that $y = ax^3 + 7$, which of the following are correct C# statements for this equation?
    a) `y = a * x * x * x + 7;`
    b) `y = a * x * x * (x + 7);`
    c) `y = (a * x) * x * (x + 7);`
    d) `y = (a * x) * x * x + 7;`
    e) `y = a * (x * x * x) + 7;`

f) y = a * x * (x * x + 7);
ANS: (a), (d), (e).

3.13   *(Order of Evaluation)* State the order of evaluation of the operators in each of the following C# statements and show the value of x after each statement is performed:

a) x = 7 + 3 * 6 / 2 - 1;
ANS: *, /, +, -, =; Value of x is 15.
b) x = 2 % 2 + 2 * 2 - 2 / 2;
ANS: %, *, /, +, -, =; Value of x is 3.
c) x = (3 * 9 * (3 + (9 * 3 / (3))));
ANS: x = (3 * 9 * (3 + (9 * 3 / (3))));
            6   4   5   3   1   2
        Value of x is 324.

3.19   What does the following code display?

```
Console.WriteLine("*\n**\n***\n****\n*****");
```

ANS:

```
*
**
***
****
*****
```

3.20   What does the following code display?

```
Console.WriteLine("*");
Console.WriteLine("***");
Console.WriteLine("*****");
Console.WriteLine("****");
Console.WriteLine("**");
```

ANS:

```
*
***
*****
****
**
```

3.21   What does the following code display?

```
Console.Write("*");
Console.Write("***");
Console.Write("*****");
Console.Write("****");
Console.WriteLine("**");
```

ANS:

```
***************
```

3.22   What does the following code display?

```
Console.Write("*");
Console.WriteLine("***");
Console.WriteLine("*****");
Console.Write("****");
Console.WriteLine("**");
```

ANS:

```
****
*****
******
```

3.23   What does the following code display?

```
string s1 = "*";
string s2 = "***";
string s3 = "*****";
Console.WriteLine($"{s1}\n{s2}\n{s3}");
```

ANS:

```
*
***
*****
```