A **self-documenting** program element is one that is self-explanatory.

An **enumeration** is a set of constants represented by identifiers.

Type-safe describes a data type for which only appropriate behaviors are allowed.

An **interactive program** is one that allows user input.

A **prompt** is an instruction to the user to enter data.

To **parse** an item is to break it into component parts.

Review Questions

1. When you use a number such as 45 in a C# program, the number is a _____.
   a. figurative constant
   b. literal constant
   c. literal variable
   d. figurative variable
2. A variable declaration must contain all of the following *except* a(n) _____.
   a. data type
   b. identifier
   c. assigned value
   d. ending semicolon
3. Which of the following is true of variable declarations?
   a. Two variables of different types can be declared in the same statement.
   b. Two variables of the same type can be declared in the same statement.
   c. Two variables of the same type must be declared in the same statement.
   d. Two variables of the same type cannot coexist in a program.
4. Assume that you have two variables declared as `int var1 = 3;` and `int var2 = 8;` Which of the following would display 838?
   a. `WriteLine("{0}{1}{2}", var1, var2);`
   b. `WriteLine("{0}{1}{0}", var2, var1);`
   c. `WriteLine("{0}{1}{2}", var2, var1);`
   d. `WriteLine("{0}{1}{0}", var1, var2);`
5. Assume that you have a variable declared as `int var1 = 3;`. Which of the following would display *X 3X?*
   a. `WriteLine("X{0}X", var1);`
   b. `WriteLine("X{0,2}X", varl);`
   c. `WriteLine("X{2,0}X", varl);`
   d. `WriteLine("X{0}{2}", varl);`

6. Assume that you have a variable declared as `int var1 = 3;`. What is the value of `22 % var1`?

   a. 21
   b. 7
   c. 1
   d. 0

7. Assume that you have a variable declared as `int var1 = 3;`. What is the value of `22 / var1`?

   a. 21
   b. 7.333
   c. 7
   d. 1

8. What is the value of the expression `4 + 2 * 3`?

   a. 0
   b. 10
   c. 18
   d. 36

9. Assume that you have a variable declared as `int var1 = 3;`. If `var2 = ++var1`, what is the value of `var2`?

   a. 2
   b. 3
   c. 4
   d. 5

10. Assume that you have a variable declared as `int var1 = 3;`. If `var2 = var1++`, what is the value of `var2`?

    a. 2
    b. 3
    c. 4
    d. 5

11. A variable that can hold the two values `true` and `false` is of type _____.

    a. `char`
    b. `it`
    c. `bool`
    d. `double`

12. Which of the following is *not* a C# comparison operator?

    a. `=>`
    b. `!=`
    c. `==`
    d. `<`

13. What is the value of the expression `6 >= 7`?

    a. 0
    b. 1
    c. `true`
    d. `false`

14. Which of the following C# types *cannot* contain floating-point numbers?

    a. `float`
    b. `double`
    c. `decimal`
    d. `int`

15. Assume that you have declared a variable as `double hourly = 13.00;`. What will the statement `WriteLine(hourly);` display?

    a. 13
    b. 13.0
    c. 13.00
    d. 13.000000

16. Assume that you have declared a variable as `double salary = 45000.00;`. Which of the following will display *$45,000*?

    a. `WriteLine(salary.ToString("c"));`
    b. `WriteLine(salary.ToString("c0"));`
    c. `WriteLine(salary);`
    d. two of these

17. When you perform arithmetic operations with operands of different types, such as adding an `int` and a `float`, _____.

    a. C# chooses a unifying type for the result
    b. you must choose a unifying type for the result
    c. you must provide a cast
    d. you receive an error message

18. Unicode is _____.

    a. an object-oriented language
    b. a subset of the C# language
    c. a 16-bit coding scheme
    d. another term for hexadecimal

19. Which of the following declares a variable that can hold the word *computer*?

    a. `string device = 'computer';`
    b. `string device = "computer";`
    c. `char device = 'computer';`
    d. `char device = "computer";`

20. Which of the following compares two string variables named `string1` and `string2` to determine if their contents are equal?

    a. `string1 = string2`
    b. `string1 == string2`
    c. `Equals.String(string1, string2)`
    d. two of the above

Exercises

Programming Exercises

1. What is the numeric value of each of the following expressions, as evaluated by the C# programming language?
   a. 2 + 5 * 3
   b. 9 / 4 + 10
   c. 10 / 3
   d. 21 % 10
   e. (5 – 1) * 3
   f. 37 / 5
   g. 64 % 8
   h. 5 + 2 * 4 – 3 * 4
   i. 3 * (2 + 5) / 5
   j. 28 % 5 – 2
   k. 19 / 2 / 2
   l. 28 / (2 + 4)

2. What is the value of each of the following Boolean expressions?
   a. 5 > 4
   b. 3 <= 3
   c. 2 + 4 > 5
   d. 6 == 7
   e. 2 + 4 <= 6
   f. 3 + 4 == 4 + 3
   g. 1 != 2
   h. 2 != 2
   i. –5 == 7–2
   j. 3 + 9 <= 0

3. Choose the best data type for each of the following, so that no memory storage is wasted. Give an example of a typical value that would be held by the variable, and explain why you chose the type you did.
   a. the number of years of school you have completed
   b. your final grade in this class
   c. the population of China
   d. the number of passengers on an airline flight
   e. one player's score in a Scrabble game
   f. the number of Electoral College votes received by a U.S. presidential candidate
   g. the number of days with below freezing temperatures in a winter in Miami, Florida
   h. one team's score in a Major League Baseball game

4. In this chapter, you learned that although a `double` and a `decimal` both hold floating-point numbers, a `double` can hold a larger value. Write a C# program named **DoubleDecimalTest** that declares and displays two variables—a `double` and a `decimal`. Experiment by assigning the same constant value to each variable so that the assignment to the `double` is legal but the assignment to the `decimal` is not. In other words, when you leave the `decimal` assignment statement in the program, an

error message should be generated that indicates the value is outside the range of the type `decimal`, but when you comment out the `decimal` assignment and its output statement, the program should compile correctly.

5. Write a C# program named **InchesToCentmeters** that declares a named constant that holds the number of centimeters in an inch: 2.54. Also declare a variable to represent a measurement in inches, and assign a value. Display the measurement in both inches and centimeters—for example, *3 inches is 7.62 centimeters*.

6. Convert the InchesToCentimeters program to an interactive application named **InchesToCentimetersInteractive**. Instead of assigning a value to the `inches` variable, accept the value from the user as input.

7. Write a C# program named **ProjectedRaises** that includes a named constant representing next year's anticipated 4 percent raise for each employee in a company. Also declare variables to represent the current salaries for three employees. Assign values to the variables, and display, with explanatory text, next year's salary for each employee.

8. Convert the `ProjectedRaises` class to an interactive application named **ProjectedRaisesInteractive**. Instead of assigning values to the salaries, accept them from the user as input.

9. Malcolm Movers charges a base rate of $200 per move plus $150 per hour and $2 per mile. Write a program named **MoveEstimator** that prompts a user for and accepts estimates for the number of hours for a job and the number of miles involved in the move and displays the total moving fee.

10. Write a program named **HoursAndMinutes** that declares a `minutes` variable to represent minutes worked on a job, and assign a value to it. Display the value in hours and minutes. For example, 197 minutes becomes 3 hours and 17 minutes.

11. Write a program named **Eggs** that declares four variables to hold the number of eggs produced in a month by each of four chickens, and assign a value to each variable. Sum the eggs, then display the total in dozens and eggs. For example, a total of 127 eggs is 10 dozen and 7 eggs.

12. Modify the `Eggs` program to create a new one named **EggsInteractive** that prompts the user for and accepts a number of eggs for each chicken.

13. Write a program named **MakeChange** that calculates and displays the conversion of an entered number of dollars into currency denominations—twenties, tens, fives, and ones. For example, $113 is 5 twenties, 1 ten, 0 fives, and 3 ones.

14. Write a program named **TestsInteractive** that prompts a user for eight test scores and displays the average of the test scores to two decimal places.

15. Write a program named **FahrenheitToCelsius** that accepts a temperature in Fahrenheit from a user and converts it to Celsius by subtracting 32 from the Fahrenheit value and multiplying the result by 5/9. Display both values to one decimal place.

16. Create an enumeration named `Month` that holds values for the months of the year, starting with `JANUARY` equal to 1. Write a program named **MonthNames** that prompts the user for a month integer. Convert the user's entry to a `Month` value, and display it.

17. Create an enumeration named `Planet` that holds the names for the eight planets in our solar system, starting with `MERCURY` and ending with `NEPTUNE`. Write a program named **Planets** that prompts the user for a numeric position, and display the name of the planet that is in the requested position.

18. Pig Latin is a nonsense language. To create a word in pig Latin, you remove the first letter and then add the first letter and "ay" at the end of the word. For example, "dog" becomes "ogday" and "cat" becomes "atcay." Write a program named **PigLatin** that allows the user to enter a word and displays the pig Latin version.

Debugging Exercises

1. Each of the following files in the Chapter.02 folder of your downloadable student files has syntax and/or logical errors. In each case, determine the problem and fix the program. After you correct the errors, save each file using the same filename preceded with *Fixed*. For example, DebugTwo1.cs will become FixedDebugTwo1.cs.

   a. DebugTwo1.cs
   b. DebugTwo2.cs
   c. DebugTwo3.cs
   d. DebugTwo4.cs

Case Problems

1. In Chapter 1, you created two programs to display the motto for the Greenville Idol competition that is held each summer during the Greenville County Fair. Now write a program named **GreenvilleRevenue** that prompts a user for the number of contestants entered in last year's competition and in this year's competition. Display all the input data. Compute and display the revenue expected for this year's competition if each contestant pays a $25 entrance fee. Also display a statement that indicates whether this year's competition has more contestants than last year's.

2. In Chapter 1, you created two programs to display the motto for Marshall's Murals. Now write a program named **MarshallsRevenue** that prompts a user for the number of interior and exterior murals scheduled to be painted during the next month. Compute the expected revenue for each type of mural. Interior murals cost $500 each, and exterior murals cost $750 each. Also display the total expected revenue and a statement that indicates whether more interior murals are scheduled than exterior ones.