22. The three main kinds of program errors are syntax errors, run-time errors, and logic errors.

23. The compiler detects syntax errors. There are other errors that are not technically syntax errors that we are lumping with syntax errors. You will learn about these later.

24. A syntax error.

25. A syntax error.

26. The text states that you should take warnings as if they had been reported as errors. You should ask your instructor for the local rules on how to handle warnings.

27. A logic error.

PRACTICE PROGRAMS

Practice Programs can generally be solved with a short program that directly applies the programming principles presented in this chapter.

1. Using your text editor, enter (that is, type in) the C++ program shown in Display 1.8. Be certain to type the first line exactly as shown in Display 1.8. In particular, be sure that the first line begins at the left-hand end of the line with no space before or after the # symbol. Compile and run the program. If the compiler gives you an error message, correct the program and recompile the program. Do this until the compiler gives no error messages. Then run your program.

2. Modify the C++ program you entered in Practice Program 1. Change the program so that it first writes the word `Hello` to the screen and then goes on to do the same things that the program in Display 1.8 does. You will only have to add one line to the program to make this happen. Recompile the changed program and run it. Then change the program even more. Add one more line that will make the program write the word `Good-bye` to the screen at the end of the program. Be certain to add the symbols `\n` to the last output statement so that it reads as follows:

   ```
   cout << "Good-bye\n";
   ```

   (Some systems require that final `\n`, and your system may be one of them.) Recompile and run the changed program.

3. Further modify the C++ program that you already modified in Practice Program 2. Change the multiplication sign * in your C++ program to a division sign /. Recompile the changed program. Run the program. Enter a 0 input for "number of peas in a pod." Notice the run-time error message due to division by zero.

4. Modify the C++ program that you entered in Practice Program 1. Change the multiplication sign * in your C++ program to an addition sign +. Recompile and run the changed program. Notice that the program compiles and runs perfectly fine, but the output is incorrect. That is because this modification is a logic error.

5. Modify the C++ program that you entered in Practice Program 1. In this version calculate the total length of fence you would need to enclose a rectangular area that is `width` feet long and `height` feet tall. The program should have variables for `width` and `height` with values entered by the user. Create another variable, `totalLength`, that stores the total length of fence needed (which your program should calculate). Output the total with an appropriate message.

VideoNote
Solution to Practice Program 1.6

6. The purpose of this exercise is to produce a catalog of typical syntax errors and error messages that will be encountered by a beginner and to continue acquainting you with the programming environment. This exercise should leave you with a knowledge of what error to look for when given any of a number of common error messages.

   Your instructor may have a program for you to use for this exercise. If not, you should use a program from one of the previous Practice Programs.

   Deliberately introduce errors to the program, compile, record the error and the error message, fix the error, compile again (to be sure you have the program corrected), then introduce another error. Keep the catalog of errors and add program errors and messages to it as you continue through this course.

   The sequence of suggested errors to introduce is:

   a Put an extra space between the < and the iostream file name.
   b. Omit one of the `<` or `>` symbols in the include directive.
   c. Omit the `int` from `int main()`.
   d. Omit or misspell the word main.
   e. Omit one of the `()`; then omit both the `()`.
   f. Continue in this fashion, deliberately misspelling identifiers (`cout`, `cin`, and so on). Omit one or both of the << in the `cout` statement; leave off the ending curly brace `}`.

PROGRAMMING PROJECTS

Programming Projects require more problem-solving than Practice Programs and can usually be solved many different ways. Visit www.myprogramminglab.com *to complete many of these Programming Projects online and get instant feedback.*

1. Write a C++ program that reads in two integers and then outputs both their sum and their product. One way to proceed is to start with the

program in Display 1.8 and to then modify that program to produce the program for this project. Be certain to type the first line of your program exactly the same as the first line in Display 1.8. In particular, be sure that the first line begins at the left-hand end of the line with no space before or after the # symbol. Also, be certain to add the symbols `\n` to the last output statement in your program. For example, the last output statement might be the following:

```
cout << "This is the end of the program.\n";
```

(Some systems require that final `\n`, and your system may be one of these.)

2. Write a program that prints out "C S !" in large block letters inside a border of *s followed by two blank lines then the message Computer Science is Cool Stuff. The output should look as follows:

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

           C C C             S S S S        !!
         C       C         S         S      !!
        C                 S                 !!
       C                   S                !!
       C                    S S S S         !!
       C                             S      !!
        C                             S     !!
         C       C         S         S
           C C C             S S S S        00

**************************************************

    Computer Science is Cool Stuff!!!
```

3. Write a program that allows the user to enter a number of quarters, dimes, and nickels and then outputs the monetary value of the coins in cents. For example, if the user enters 2 for the number of quarters, 3 for the number of dimes, and 1 for the number of nickels, then the program should output that the coins are worth 85 cents.

VideoNote
Solution to Programming Project 1.3

4. Write a program that allows the user to enter a time in seconds and then outputs how far an object would drop if it is in freefall for that length of time. Assume that the object starts at rest, there is no friction or resistance from air, and there is a constant acceleration of 32 feet per second due to gravity. Use the equation:

$$\text{distance} = \frac{\text{acceleration} \times \text{time}^2}{2}$$

You should first compute the product and then divide the result by 2. (The reason for this will be discussed later in the book.)

5. Write a program that inputs a character from the keyboard and then outputs a large block letter "C" composed of that character. For example, if the user inputs the character "X," then the output should look as follows:

```
  X X X
 X     X
X
X
X
X
X
 X     X
  X X X
```