```
12     cout <<                          ; // prompt user for data
13     cin >> x >> y >> z; // read three integers from user
14     result = x * y * z; // multiply the three integers; store result
15     cout <<                          << result << endl; // print result; end line
16  } // end function main
```

2.6 a) *Error:* Semicolon after the right parenthesis of the condition in the `if` statement.
Correction: Remove the semicolon after the right parenthesis. [*Note:* The result of this error is that the output statement executes whether or not the condition in the `if` statement is true.] The semicolon after the right parenthesis is an empty statement that does nothing. We'll say more about the empty statement in Chapter 4.

b) *Error:* The incorrect relational operator `=>`.
Correction: Change `=>` to `>=`, and you may want to change "equal to or greater than" to "greater than or equal to" as well.

Exercises

2.7 Discuss the meaning of each of the following objects:

a) `std::cin`

b) `std::cout`

2.8 Fill in the blanks in each of the following:

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

b) The object used to print information on the screen is _____.

c) A C++ statement that makes a decision is _____.

d) Most calculations are normally performed by _____ statements.

e) The _____ object inputs values from the keyboard.

2.9 Write a single C++ statement or line that accomplishes each of the following:

a) Print the message `"Enter two numbers"`.

b) Assign the product of variables `b` and `c` to variable `a`.

c) State that a program performs a payroll calculation (i.e., use text that helps to document a program).

d) Input three integer values from the keyboard into integer variables `a`, `b` and `c`.

2.10 State which of the following are *true* and which are *false*. If *false*, explain your answers.

a) All operators are evaluated from left to right.

b) The following are all valid variable names: `_under_bar_`, `m928134`, `t5`, `j7`, `her_sales`, `his_account_total`, `a`, `b`, `c`, `z`, `z2`.

c) The statement `cout << "a = 5;";` is a typical example of an assignment statement.

d) A valid arithmetic expression with no parentheses is evaluated from left to right.

e) The following are all invalid variable names: `3g`, `87`, `67h2`, `h22`, `2h`.

2.11 Fill in the blanks in each of the following:

a) What arithmetic operations are on the same level of precedence as multiplication? _____.

b) When parentheses are nested, which set of parentheses is evaluated first in an arithmetic expression? _____.

c) A location in the computer's memory that may contain different values at various times throughout the execution of a program is called a(n) _____.

2.12 What, if anything, prints when each of the following statements is performed? If nothing prints, then answer "nothing." Assume `x = 2` and `y = 3`.

a) `cout << x;`

b) `cout << x + x;`
c) `cout <<  ;`
d) `cout <<  << x;`
e) `cout << x + y <<  << y + x;`
f) `z = x + y;`
g) `cin >> x >> y;`
h) `cout << "  " << ;`
i) `cout <<  ;`

2.13 Which of the following statements contain variables whose values are replaced?
a) `cin >> b >> c >> d >> e >> f;`
b) `p = i + j + k + 7;`
c) `cout <<  ;`
d) `cout <<  ;`

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

2.15 *(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 =  + * / - ;`
b) `x =  % + * - / ;`
c) `x  ( * * ( + ( * / ( ))));`

2.16 *(Arithmetic)* Write a program that asks the user to enter two numbers, obtains the two numbers from the user and prints the sum, product, difference, and quotient of the two numbers.

2.17 *(Printing)* Write a program that prints the numbers 1 to 4 on the same line with each pair of adjacent numbers separated by one space. Do this several ways:
a) Using one statement with one stream insertion operator.
b) Using one statement with four stream insertion operators.
c) Using four statements.

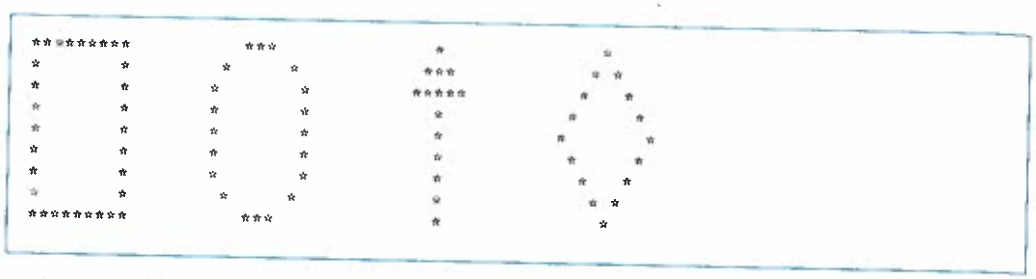
2.18 *(Comparing Integers)* Write a program that asks the user to enter two integers, obtains the numbers from the user, then prints the larger number followed by the words "`is larger.`" If the numbers are equal, print the message "`These numbers are equal.`"

2.19 *(Arithmetic, Smallest and Largest)* Write a program that inputs three integers from the keyboard and prints the sum, average, product, smallest and largest of these numbers. The screen dialog should appear as follows:

```
Input three different integers: 13 27 14
Sum is 54
Average is 18
Product is 4914
Smallest is 13
Largest is 27
```

2.20 ***(Diameter, Circumference and Area of a Circle)*** Write a program that reads in the radius of a circle as an integer and prints the circle's diameter, circumference and area. Use the constant value 3.14159 for π. Do all calculations in output statements. [*Note:* In this chapter, we've discussed only integer constants and variables. In Chapter 4 we discuss floating-point numbers, i.e., values that have decimal points.]

2.21 ***(Displaying Shapes with Asterisks)*** Write a program that prints a box, an oval, an arrow and a diamond as follows:



2.22 What does the following code print?

```
cout <<                                    << endl;
```

2.23 ***(Largest and Smallest Integers)*** Write a program that reads in five integers and determines and prints the largest and the smallest integers in the group. Use only the programming techniques you learned in this chapter.

2.24 ***(Odd or Even)*** Write a program that reads an integer and determines and prints whether it's odd or even. [*Hint:* Use the remainder operator (%). An even number is a multiple of two. Any multiple of 2 leaves a remainder of zero when divided by 2.]

2.25 ***(Multiples)*** Write a program that reads in two integers and determines and prints if the first is a multiple of the second. [*Hint:* Use the remainder operator (%).]

2.26 ***(Checkerboard Pattern)*** Display the following checkerboard pattern with eight output statements, then display the same pattern using as few statements as possible.

2.27 ***(Integer Equivalent of a Character)*** Here is a peek ahead. In this chapter you learned about integers and the type `int`. C++ can also represent uppercase letters, lowercase letters and a considerable variety of special symbols. C++ uses small integers internally to represent each different character. The set of characters a computer uses and the corresponding integer representations for those characters are called that computer's character set. You can print a character by enclosing that character in single quotes, as with

```
cout <<     ; // print an uppercase A
```

You can print the integer equivalent of a character using `static_cast` as follows:

```
cout << static_cast<int>(   ); // print 'A' as an integer
```

This is called a **cast** operation (we formally introduce casts in Chapter 4). When the preceding statement executes, it prints the value 65 (on systems that use the **ASCII character set**). Write a program that prints the integer equivalent of a character typed at the keyboard. Store the input in a variable of type char. Test your program several times using uppercase letters, lowercase letters, digits and special characters (such as $).

2.28 *(Digits of an Integer)* Write a program that inputs a five-digit integer, separates the integer into its digits and prints them separated by three spaces each. [*Hint:* Use the integer division and remainder operators.] For example, if the user types in 42339, the program should print:

```
4   2   3   3   9
```

2.29 *(Table)* Using the techniques of this chapter, write a program that calculates the squares and cubes of the integers from 0 to 10. Use tabs to print the following neatly formatted table of values:

```
integer square  cube
0       0       0
1       1       1
2       4       8
3       9       27
4       16      64
5       25      125
6       36      216
7       49      343
8       64      512
9       81      729
10      100     1000
```

Making a Difference

2.30 *(Body Mass Index Calculator)* We introduced the body mass index (BMI) calculator in Exercise 1.9. The formulas for calculating BMI are

$$BMI = \frac{weightInPounds \times 703}{heightInInches \times heightInInches}$$

or

$$BMI = \frac{weightInKilograms}{heightInMeters \times heightInMeters}$$

Create a BMI calculator application that reads the user's weight in pounds and height in inches (or, if you prefer, the user's weight in kilograms and height in meters), then calculates and displays the user's body mass index. Also, the application should display the following information from the Department of Health and Human Services/National Institutes of Health so the user can evaluate his/her BMI:

```
BMI VALUES
Underweight: less than 18.5
Normal:      between 18.5 and 24.9
Overweight:  between 25 and 29.9
Obese:       30 or greater
```