2

Introduction to C++ Programming: Solutions

What's in a name? that which
we call a rose
By any other name would smell
as sweet.

—William Shakespeare

When faced with a decision, I always ask, "What would be the most fun?"

—Peggy Walker

"Take some more tea," the March Hare said to Alice, very earnestly. "I've had nothing yet," Alice replied in an offended tone: "so I can't take more." "You mean you can't take less," said the Hatter: "it's very easy to take more than nothing."

—Lewis Carroll

High thoughts must have high language.

—Aristophane

Objectives

In this chapter you'll learn:

- To write simple computer programs in C++.
- To write simple input and output statements.
- To use fundamental types.
- Basic computer memory concepts.
- To use arithmetic operators.
- The precedence of arithmetic operators.
- To write simple decision-making statements.

Self-Review Exercises

2.1 Fill in the blanks in each of the following.

a) Every C++ program begins execution at the function ________.

ANS: `main`.

b) A ________ begins the body of every function and a ________ ends the body.

ANS: left brace (`{`), right brace (`}`).

c) Every C++ statement ends with a(n) ________.

ANS: semicolon.

d) The escape sequence `\n` represents the ________ character, which causes the cursor to position to the beginning of the next line on the screen.

ANS: newline.

e) The ________ statement is used to make decisions.

ANS: `if`.

2.2 State whether each of the following is *true* or *false*. If *false*, explain why. Assume the statement `using std::cout;` is used.

a) Comments cause the computer to print the text after the `//` on the screen when the program is executed.

ANS: False. Comments do not cause any action to be performed when the program is executed. They are used to document programs and improve their readability.

b) The escape sequence `\n`, when output with `cout` and the stream insertion operator, causes the cursor to position to the beginning of the next line on the screen.

ANS: True.

c) All variables must be declared before they are used.

ANS: True.

d) All variables must be given a type when they are declared.

ANS: True.

e) C++ considers the variables `number` and `NuMbEr` to be identical.

ANS: False. C++ is case sensitive, so these variables are unique.

f) Declarations can appear almost anywhere in the body of a C++ function.

ANS: True.

g) The modulus operator (`%`) can be used only with integer operands.

ANS: True.

h) The arithmetic operators `*`, `/`, `%`, `+` and – all have the same level of precedence.

ANS: False. The operators `*`, `/` and `%` have the same precedence, and the operators `+` and `-` have a lower precedence.

i) A C++ program that prints three lines of output must contain three statements using `cout` and the stream insertion operator.

ANS: False. A single statement using `cout` containing multiple `\n` escape sequences also can print several lines.

2.3 Write a *single* C++ statement to accomplish each of the following (assume that `using` declarations have *not* been used):

a) Declare the variables `c`, `thisIsAVariable`, `q76354` and `number` to be of type `int`.

ANS: `int c, thisIsAVariable, q76354, number;`

b) Prompt the user to enter an integer. End your prompting message with a colon (`:`) followed by a space and leave the cursor positioned after the space.

ANS: `std::cout << "Enter an integer: ";`

c) Read an integer from the user at the keyboard and store it in integer variable `age`.

ANS: `std::cin >> age;`

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

ANS:
```
if ( number != 7 )
   std::cout << "The variable number is not equal to 7\n";
```

e) Print the message `"This is a C++ program"` on one line.

ANS: `std::cout << "This is a C++ program\n";`

f) Print the message `"This is a C++ program"` on two lines. End the first line with `C++`.

ANS: `std::cout << "This is a C++\nprogram\n";`

g) Print the message `"This is a C++ program"` with each word on a separate line.

ANS: `std::cout << "This\nis\na\nC++\nprogram\n";`

h) Print the message `"This is a C++ program"`. Separate each word from the next by a tab.

ANS: `std::cout << "This\tis\ta\tC++\tprogram\n";`

2.4 Write a statement (or comment) to accomplish each of the following (assume that `using` declarations have been used):

a) State that a program calculates the product of three integers.

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

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

ANS:
```
int x;
int y;
int z;
int result;
```

c) Prompt the user to enter three integers.

ANS: `cout << "Enter three integers: ";`

d) Read three integers from the keyboard and store them in the variables `x`, `y` and `z`.

ANS: `cin >> x >> y >> z;`

e) 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;`

f) Print `"The product is "` followed by the value of the variable `result`.

ANS: `cout << "The product is " << result << endl;`

g) Return a value from `main` indicating that the program terminated successfully.

ANS: `return 0;`

2.5 Using the statements you wrote in Exercise 2.4, write a complete program that calculates and displays the product of three integers. Add comments to the code where appropriate. [*Note:* You'll need to write the necessary `using` declarations.]

ANS:

```
// Calculate the product of three integers
#include <iostream> // allows program to perform input and output
using namespace std;

// function main begins program execution
int main()
{
   int x; // first integer to multiply
   int y; // second integer to multiply
   int z; // third integer to multiply
   int result; // the product of the three integers

   cout << "Enter three integers: "; // prompt user for data
   cin >> x >> y >> z; // read three integers from user
   result = x * y * z; // multiply the three integers; store result
```

```
    cout << "The product is " << result << endl; // print result; end line
} // end function main
```

```
Enter three integers: 2 4 6
The product is 48
```

2.6 Identify and correct the errors in each of the following statements (assume that the statement `using std::cout;` is used):

a) `if ( c < 7 );`
```
    cout << "c is less than 7\n";
```

ANS: 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 will be executed whether or not the condition in the `if` statement is true.] The semicolon after the right parenthesis is a null (or empty) statement—a statement that does nothing. We'll learn more about the null statement in the next chapter.

b) `if ( c => 7 )`
```
    cout << "c is equal to or greater than 7\n";
```

ANS: Error: The 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`

ANS: This object refers to the standard input device that is normally connected to the keyboard.

b) `std::cout`

ANS: This object refers to the standard output device that is normally connected to the screen.

2.8 Fill in the blanks in each of the following:

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

ANS: Comments

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

ANS: `cout`

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

ANS: `if`

d) Most calculations are normally performed by ________ statements.

ANS: assignment

e) The ________ object inputs values from the keyboard.

ANS: `cin`

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

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

ANS: `cout << "Enter two numbers";`

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

ANS: `a = b * c;`

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

ANS: `// Payroll calculation program`

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

ANS: `cin >> a >> b >> c;`

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

a) C++ operators are evaluated from left to right.

ANS: False. Some operators are evaluated from left to right, while other operators are evaluated 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`, `z2`.

ANS: True.

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

ANS: False. The statement is an output statement. The text `a = 5;` is output to the screen.

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

ANS: False. Arithmetic operators can appear in any order in an expression, so the expression is `a = b + c * d;` actually evaluates from right to left because of the rules of operator precedence.

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

ANS: False. `h22` is a valid variable name. The others are invalid because they each begin with a digit.

2.11 Fill in the blanks in each of the following:

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

ANS: division and modulus.

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

ANS: innermost.

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 ________.

ANS: variable.

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

a) `cout << x;`

ANS: `2`

b) `cout << x + x;`

ANS: `4`

c) `cout << "x=";`

ANS: `x=`

d) `cout << "x = " << x;`

ANS: `x = 2`

e) `cout << x + y << " = " << y + x;`

ANS: `5 = 5`

f) `z = x + y;`

ANS: nothing.

g) `cin >> x >> y;`

ANS: nothing.

h) `// cout << "x + y = " << x + y;`

ANS: nothing (because it is a comment).

i) `cout << "\n";`

ANS: A newline is output which positions the cursor at the beginning of the next line on the screen.

2.13 Which of the following C++ statements contain variables whose values are replaced?

a) `cin >> b >> c >> d >> e >> f;`

b) `p = i + j + k + 7;`

c) `cout << "variables whose values are replaced";`

d) `cout << "a = 5";`

ANS: Parts (a) and (b).

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 + 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: Parts (a), (d) and (e).

2.15 *(Order of Evalution)* 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: `*, /, +, -, =, 15`

b) `x = 2 % 2 + 2 * 2 - 2 / 2;`

ANS: `%, *, /, +, -, =, 3`

c) `x = ( 3 * 9 * ( 3 + ( 9 * 3 / ( 3 ) ) ) );`

ANS: innermost parentheses around 3, *, /, +, *, *, `324`

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.

ANS:

```cpp
// Exercise 2.16 Solution: ex02_16.cpp
#include <iostream> // allows program to perform input and output
using namespace std; // program uses names from the std namespace

int main()
{
   int number1; // first integer read from user
   int number2; // second integer read from user

   cout << "Enter two integers: "; // prompt user for data
   cin >> number1 >> number2; // read values from user

   // output the results
   cout << "The sum is " << number1 + number2
      << "\nThe product is " << number1 * number2
      << "\nThe difference is " << number1 - number2
      << "\nThe quotient is " << number1 / number2 << endl;
} // end main
```

```
Enter two integers: 8 22
The sum is 30
The product is 176
The difference is -14
The quotient is 0
```

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.

ANS:

```cpp
// Exercise 2.17 Solution: ex02_17.cpp
#include <iostream> // allows program to perform input and output
using namespace std; // program uses names from the std namespace

int main()
{
   // Part a
   cout << "1 2 3 4\n";

   // Part b
   cout << "1 " << "2 " << "3 " << "4\n";

   // Part c
   cout << "1 ";
   cout << "2 ";
   cout << "3 ";
   cout << "4" << endl;
} // end main
```

```
1 2 3 4
1 2 3 4
1 2 3 4
```

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."

ANS:

```cpp
// Exercise 2.18 Solution: ex02_18.cpp
#include <iostream> // allows program to perform input and output
using namespace std;

int main()
{
   int number1; // first integer read from user
   int number2; // second integer read from user

   cout << "Enter two integers: "; // prompt user for data
   cin >> number1 >> number2; // read two integers from user

   if ( number1 == number2 )
      cout << "These numbers are equal." << endl;

   if ( number1 > number2 )
      cout << number1 << " is larger." << endl;

   if ( number2 > number1 )
      cout << number2 << " is larger." << endl;
} // end main
```

```
Enter two integers: 22 8
22 is larger.
```

```
Enter two integers: 8 22
22 is larger.
```

```
Enter two integers: 22 22
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
```

ANS:

```cpp
// Exercise 2.19 Solution: ex02_19.cpp
#include <iostream> // allows program to perform input and output
using namespace std; // program uses names from the std namespace

int main()
{
   int number1; // first integer read from user
   int number2; // second integer read from user
   int number3; // third integer read from user
   int smallest; // smallest integer read from user
   int largest; // largest integer read from user

   cout << "Input three different integers: "; // prompt
   cin >> number1 >> number2 >> number3; // read three integers

   largest = number1; // assume first integer is largest

   if ( number2 > largest ) // is number2 larger?
      largest = number2; // number2 is now the largest

   if ( number3 > largest ) // is number3 larger?
      largest = number3; // number3 is now the largest

   smallest = number1; // assume first integer is smallest

   if ( number2 < smallest ) // is number2 smaller?
      smallest = number2; // number2 is now the smallest

   if ( number3 < smallest ) // is number3 smaller?
      smallest = number3; // number3 is now the smallest

   cout << "Sum is " << number1 + number2 + number3
      << "\nAverage is " << ( number1 + number2 + number3 ) / 3
      << "\nProduct is " << number1 * number2 * number3
      << "\nSmallest is " << smallest
      << "\nLargest is " << largest << endl;
} // end main
```

```
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 have discussed only integer constants and variables. In Chapter 3 we discuss floating-point numbers, i.e., values that can have decimal points.]

ANS:

```
// Exercise 2.20 Solution: ex02_20.cpp
#include <iostream> // allows program to perform input and output
using namespace std; // program uses names from the std namespace

int main()
{
   int radius; // variable to store a circle's radius

   cout << "Enter the circle radius: "; // prompt user for radius
   cin >> radius; // read radius from user

   cout << "Diameter is " << radius * 2.0
      << "\nCircumference is " << 2 * 3.14159 * radius
      << "\nArea is " << 3.14159 * radius * radius << endl;
} // end main
```

```
Enter the circle radius: 8
Diameter is 16
Circumference is 50.2654
Area is 201.062
```

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

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

ANS:

```
// Exercise 2.21 Solution: ex02_21.cpp
#include <iostream> // allows program to perform input and output
using namespace std; // program uses names from the std namespace

int main()
{
   cout << "*********      ***         *         *\n"
        << "*       *    *     *      ***       * *\n"
        << "*       *   *       *    *****     *   *\n"
        << "*       *   *       *      *      *     *\n"
        << "*       *   *       *      *     *       *\n"
        << "*       *   *       *      *      *     *\n"
        << "*       *   *       *      *       *   *\n"
        << "*       *    *     *       *        * *\n"
        << "*********      ***         *         *" << endl;
} // end main
```

2.22 What does the following code print?

```
cout << "*\n**\n***\n****\n*****" << endl;
```

ANS:

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

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.

ANS:

```
// Exercise 2.23 Solution: ex02_23.cpp
#include <iostream> // allows program to perform input and output
using namespace std; // program uses names from the std namespace

int main()
{
   int number1; // first integer read from user
   int number2; // second integer read from user
   int number3; // third integer read from user
   int number4; // fourth integer read from user
   int number5; // fifth integer read from user
   int smallest; // smallest integer read from user
   int largest; // largest integer read from user

   cout << "Enter five integers: "; // prompt user for data
   cin >> number1 >> number2 >> number3 >> number4 >> number5;

   largest = number1; // assume first integer is largest
   smallest = number1; // assume first integer is smallest

   if ( number2 > largest ) // is number2 larger?
      largest = number2; // number2 is new largest

   if ( number3 > largest ) // is number3 larger?
      largest = number3; // number3 is new largest

   if ( number4 > largest ) // is number4 larger?
      largest = number4; // number4 is new largest

   if ( number5 > largest ) // is number5 larger?
      largest = number5; // number5 is new largest

   if ( number2 < smallest ) // is number2 smaller?
      smallest = number2; // number2 is new smallest

   if ( number3 < smallest ) // is number3 smaller?
      smallest = number3; // number3 is new smallest

   if ( number4 < smallest ) // is number4 smaller?
      smallest = number4; // number4 is new smallest

   if ( number5 < smallest ) // is number5 smaller?
      smallest = number5; // number5 is new smallest

   cout << "Largest is " << largest
      << "\nSmallest is " << smallest << endl;
} // end main
```

```
Enter five integers: 88 22 8 78 21
Largest is 88
Smallest is 8
```

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

ANS:

```
// Exercise 2.24 Solution: ex02_24.cpp
#include <iostream> // allows program to perform input and output
using namespace std; // program uses names from the std namespace

int main()
{
   int number; // integer read from user

   cout << "Enter an integer: "; // prompt
   cin >> number; // read integer from user

   if ( number % 2 == 0 )
      cout << "The integer " << number << " is even." << endl;

   if ( number % 2 != 0 )
      cout << "The integer " << number << " is odd." << endl;
} // end main
```

```
Enter an integer: 73
The integer 73 is odd.
```

```
Enter an integer: 74
The integer 74 is even.
```

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 modulus operator.]

ANS:

```
// Exercise 2.25 Solution: ex02_25.cpp
#include <iostream> // allows program to perform input and output
using namespace std;

int main()
{
   int number1; // first integer read from user
   int number2; // second integer read from user

   cout << "Enter two integers: "; // prompt
   cin >> number1 >> number2; // read two integers from user

   // using modulus operator
   if ( number1 % number2 == 0 )
      cout << number1 << " is a multiple of " << number2 << endl;

   if ( number1 % number2 != 0 )
      cout << number1 << " is not a multiple of " << number2 << endl;
} // end main
```

```
Enter two integers: 22 8
22 is not a multiple of 8
```

```
Enter two integers: 24 8
24 is a multiple of 8
```

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

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

ANS:

```
// Exercise 2.26 Solution: ex02_26.cpp
#include <iostream> // allows program to perform input and output
using namespace std; // program uses names from the std namespace
```

```

int main()
{
   // Eight output statements
   cout << "* * * * * * * *\n";
   cout << " * * * * * * * *\n";
   cout << "* * * * * * * *\n";
   cout << " * * * * * * * *\n";
   cout << "* * * * * * * *\n";
   cout << " * * * * * * * *\n";
   cout << "* * * * * * * *\n";
   cout << " * * * * * * * *\n\n";

   // One output statement; 3 parts
   cout << "* * * * * * * *\n * * * * * * * *\n* * * * * * * *\n"
      << " * * * * * * * *\n* * * * * * * *\n * * * * * * * *\n"
      << "* * * * * * * *\n * * * * * * * *\n";

   cout << endl; // ensure everything is displayed
} // end main
```

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

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

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

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

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

This is called a **cast** operation (we formally introduce casts in Chapter 3). 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 (like $).

ANS:

```
// Exercise 2.27 Solution: ex02_27.cpp
#include <iostream> // allows program to perform input and output
using namespace std; // program uses names from the std namespace

int main()
{
   char symbol; // char read from user

   cout << "Enter a character: "; // prompt user for data
   cin >> symbol; // read the character from the keyboard

   cout << symbol << "'s integer equivalent is "
      << static_cast< int >( symbol ) << endl;
} // end main
```

```
Enter a character: A
A's integer equivalent is 65
```

```
Enter a character: B
B's integer equivalent is 66
```

```
Enter a character: a
a's integer equivalent is 97
```

```
Enter a character: 7
7's integer equivalent is 55
```

```
Enter a character: $
$'s integer equivalent is 36
```

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

```
4   2   3   3   9
```

ANS:

```cpp
// Exercise 2.28 Solution: ex02_28.cpp
#include <iostream> // allows program to perform input and output
using namespace std; // program uses names from the std namespace

int main()
{
   int number; // integer read from user

   cout << "Enter a five-digit integer: "; // prompt
   cin >> number; // read integer from user

   cout << number / 10000 << "   ";
   number = number % 10000;
   cout << number / 1000 << "   ";
   number = number % 1000;
   cout << number / 100 << "   ";
   number = number % 100;
   cout << number / 10 << "   ";
   number = number % 10;
   cout << number << endl;
} // end main
```

```
Enter a five-digit integer: 42339
4   2   3   3   9
```

2.29 *(Table)* Using only the techniques you learned in this chapter, write a program that calculates the squares and cubes of the integers from 0 to 10 and uses 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
```

ANS:

```cpp
// Exercise 2.29 Solution: ex02_29.cpp
#include <iostream> // allows program to perform input and output
using namespace std;

int main()
{
   int number; // integer to square and cube

   number = 0; // set number to 0
   cout << "integer\tsquare\tcube\n"; // output column heads

   // output the integer, its square and its cube
   cout << number << '\t' << number * number << '\t'
      << number * number * number << "\n";

   number = 1; // set number to 1
   cout << number << '\t' << number * number << '\t'
      << number * number * number << "\n";

   number = 2; // set number to 2
   cout << number << '\t' << number * number << '\t'
      << number * number * number << "\n";

   number = 3; // set number to 3
   cout << number << '\t' << number * number << '\t'
      << number * number * number << "\n";

   number = 4; // set number to 4
   cout << number << '\t' << number * number << '\t'
      << number * number * number << "\n";

   number = 5; // set number to 5
   cout << number << '\t' << number * number << '\t'
      << number * number * number << "\n";

   number = 6; // set number to 6
   cout << number << '\t' << number * number << '\t'
      << number * number * number << "\n";

   number = 7; // set number to 7
   cout << number << '\t' << number * number << '\t'
      << number * number * number << "\n";

   number = 8; // set number to 8
   cout << number << '\t' << number * number << '\t'
      << number * number * number << "\n";

   number = 9; // set number to 9
   cout << number << '\t' << number * number << '\t'
      << number * number * number << "\n";

   number = 10; // set number to 10
   cout << number << '\t' << number * number << '\t'
      << number * number * number << endl;
} // end main
```

```
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 1.13. 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
```

[*Note:* In this chapter, you learned to use the `int` type to represent whole numbers. The BMI calculations when done with `int` values will both produce whole-number results. In Chapter 3 you'll learn to use the `double` type to represent numbers with decimal points. When the BMI calculations are performed with `double`s, they'll both produce numbers with decimal points—these are called "floating-point" numbers.]

ANS:

```cpp
// Exercise 2.30 Solution: ex02_30.cpp
#include <iostream> // allows program to perform input and output
using namespace std; // program uses names from the std namespace

int main()
{
   int weight, height; // variables to store weight and height

   // prompt the user for their weight and height and read them in
   cout << "Enter weight (lbs): ";
   cin >> weight;
   cout << "Enter height (in): ";
   cin >> height;

   // calculate bmi (rounds down due to integer division)
   int bmi = ( weight * 703 ) / ( height * height );

   cout << "\nYour BMI is: " << bmi << "\n\n"; // display user's BMI

   // display BMI information table
   cout << "BMI VALUES                        \n";
   cout << "Underweight: less than 18.5       \n";
   cout << "Normal:      between 18.5 and 24.9\n";
   cout << "Overweight:  between 25 and 29.9  \n";
   cout << "Obese:       30 or greater        \n";
} // end main
```

2.31 *(Car-Pool Savings Calculator)* Research several car-pooling websites. Create an application that calculates your daily driving cost, so that you can estimate how much money could be saved by car pooling, which also has other advantages such as reducing carbon emissions and reducing traffic congestion. The application should input the following information and display the user's cost per day of driving to work:

a) Total miles driven per day.
b) Cost per gallon of gasoline.
c) Average miles per gallon.
d) Parking fees per day.
e) Tolls per day.

ANS:

```
// Exercise 2.31 Solution: ex02_31.cpp
#include <iostream> // allows program to perform input and output
using namespace std; // program uses names from the std namespace

int main()
{
   // variables to store statistics
   int miles, cost, mpg, parking, tolls;

   // prompt user for each variable and read it in
   cout << "Enter miles driven per day: ";
   cin >> miles;
   cout << "Enter cost per gallon of gas (in cents): ";
   cin >> cost;
   cout << "Enter average miles per gallon: ";
   cin >> mpg;
   cout << "Enter parking fees per day (in dollars): ";
   cin >> parking;
   cout << "Enter tolls per day (in dollars): ";
   cin >> tolls;

   // calculate their daily cost in dollars
   int total = miles * cost / mpg / 100 + parking + tolls;

   // display cost
   cout << "\nYour daily driving cost is: " << total << endl;
} // end main
```