Note

In C++, only integer or character types may be used as control expressions in switch statements.

EXERCISE 7-8 USING switch

1. Open *SHIPPING.CPP*. The program includes the segment from Figure 7-14.
2. Compile, link, and run the program. Choose shipping method 2.
3. Add a fourth shipping option called *Carrier Pigeon* to the menu and add the necessary code to the switch structure. You decide how much it should cost to ship by carrier pigeon.
4. Compile, link, and run to test your addition to the options.
5. Save the source code as *PIGEON.CPP* and close.

Nested if/else structures could be used in the place of the switch structure. But the switch structure is easier to use and a programmer is less prone to making errors that are related to braces and indentions. Remember, however, that an integer or character data type is required in the control expression of a switch structure. Nested if's must be used if you are comparing floats.

When using character types in a switch structure, enclose the characters in single quotes like any other character literal. The following code segment is an example of using character literals in a switch structure.

```
switch(character_entered)
 {
   case 'A':
     cout << "The character entered was A, as in albatross.\n";
     break;
   case 'B':
     cout << "The character entered was B, as in butterfly.\n";
     break;
   default:
     cout << "Illegal entry\n";
     break;
 }
```

Extra for Experts

C++ allows you to place your case statements in any order. You can, however, increase the speed of your program by placing the more common choices at the top of the switch structure and less common ones toward the bottom. The reason is that the computer makes the comparisons in the order they appear in the switch structure. The sooner a match is found, the sooner the computer can move on to other processing.

SECTION 7.2 QUESTIONS

1. What is the purpose of the *break* keyword?
2. Write an if structure that prints the word *help* to the screen if the variable **need_help** is equal to 1.
3. Write an if/else structure that prints the word *Full* to the screen if the float variable **fuel_level** is equal to 1 and prints the value of **fuel_level** if it is not equal to 1.
4. What is wrong with the if structure below?

```
if (x > y);
  { cout << "x is greater than y\n"; }
```

5. What is wrong with the if structure below?

```
if (x = y)
  { cout << "x is equal to y\n"; }
```

PROBLEM 7.2.1

Write a program that asks for an integer and displays for the user whether the number is even or odd. *Hint:* Use if/else and the modulus operator. Save the source code file as *EVENODD.CPP*.

PROBLEM 7.2.2

Rewrite *FINAL.CPP* so that it begins with the assumption that the student is exempt and makes comparisons to see if the student must take the test. Save the revised source code as *FINAL2.CPP*.

KEY TERMS

bitwise operators
control expression
fuzzy logic
if structure
if/else structure
logical operators
menu
nested
one-way selection structure
relational operators
selection structures
sequence structures
short-circuit evaluation
switch structure
truth tables
two-way selection structure