CHAPTER 8

USING THE do while STATEMENT

The answers for the Using the do while Statement section are located at the end of the section.

1. Write a posttest loop that adds together the integers from 10 through 100. Use an int variable named num to keep track of the integers, and use an int variable named sum to store the sum of the integers. The num variable was initialized to 10 when it was declared. The sum variable was initialized to 0 when it was declared. Use the do while statement.

2. Write a while clause that will stop the posttest loop when the value in the inStock variable is less than or equal to the value in the reorder variable.

3. A char variable named letter contains an uppercase letter. Write a while clause that processes the posttest loop instructions as long as the variable's value is either Y or T.

4. Write a posttest loop that adds together the integers 5, 15, 25, 35, 45, 55, 65, 75, 85, and 95. Use an int variable named num to keep track of the integers. Store the sum in an int variable named sum. The num variable was initialized to 5 when it was declared. The sum variable was initialized to 0 when it was declared. Use the do while statement.

5. Write a posttest loop that displays the numbers .05, .06, .07, .08, .09, .10, .11, .12, .13, .14, and .15. Use the num variable to keep track of the numbers. The variable has the double data type and was initialized to .05 when it was created. Use the do while statement.

ANSWERS FOR THE USING THE do while STATEMENT SECTION

1.
```
do
{
     sum += num;
     num += 1;
}  while (num <= 100);
```
2.
```
} while (inStock > reorder);
```
3.
```
} while (letter == 'Y' || letter == 'T');
```
4.
```
do
{
     sum += num;
     num += 10;
}  while (num <= 95);
```
5.
```
do
{
     cout << num << endl;
     num += .01
}  while (num <= .15);
```

NESTED LOOPS

The answers for the Nested Loops section are located at the end of the section.

1. Rewrite the following code using the `while` statement.

```
for (int outer = 1; outer <= 3; outer += 1)
{
     cout << "Outer" << endl;
     for (int inner = 1; inner <= 2; inner += 1)
          cout << "Inner" << endl;
     //end for
}  //end for
```

2. Write the code to display the following pattern using two `for` statements along with the plus sign.

```
++++
+++
++
+
```

3. Rewrite the previous code using the `while` statement for the outer loop.

4. Analyze the problem specification shown in WM-Figure 8-1.

Professor Smith wants a program that allows him to enter three test scores for each of five students. The program should calculate and display each student's average score.

WM-Figure 8-1 Problem specification for Professor Smith

5. Code the algorithm corresponding to the Professor Smith problem from WM-Figure 8-1. Use the `for` statement.

6. Modify the previous code so that it allows Professor Smith to enter the test scores for an unknown number of students. Use a posttest loop for the outer repetition structure.

ANSWERS FOR THE NESTED LOOPS SECTION

1.

```
int outer = 1; int
inner = 1; while
(outer <= 3)
{
     cout << "Outer" << endl;
     while (inner <= 2)
     {
          cout << "Inner" << endl;
          inner += 1;
     }  //end while
     inner = 1;
     outer += 1;
}  //end while
```

2.
```
for (int line = 4; line >= 1; line -= 1)
{
      for (int plusSign = 1; plusSign <= line; plusSign += 1)
           cout << "+";
      //end for
      cout << endl;
} //end for
```

3.
```
int line = 4;
while (line >= 1)
{
      for (int plusSign = 1; plusSign <= line; plusSign += 1)
           cout << "+";
      //end for
      cout << endl;
      line -= 1;
} //end while
```

4.

Input	Processing	Output
score 1 score 2 score 3	Processing items: number of students (counter) number of students (counter) sum (accumulator) Algorithm: repeat for (number of students from 1 to 5 in increments of 1) assign 0 to the sum repeat for (number of test scores from 1 to 3 in increments of 1) enter a test score add the test score to the sum end repeat calculate the average score by dividing the sum by the number of test scores display the average score end repeat	average score (for each student)

WM-Figure 8-2 Solution for the Professor Smith problem

5.
```
int sum = 0;
int score = 0;
double avg = 0.0;
for (int numStudents = 1; numStudents <= 5; numStudents += 1)
{
     sum = 0
     for (int numScores = 1; numScores <= 3; numScores += 1)
     {
          cout << "Score " << numScores << ": ";
          cin >> score;
          sum += score;
     } //end for
     avg = sum / 3.0
     cout << "Average for student "
     << numStudents << ": " << avg << endl;
} //end for
```

6.
```
int sum = 0;
int score = 0;
double avg = 0.0;
int numStudents = 1;
char anotherStud = 'Y';
do
{
     sum = 0
     for (int numScores = 1; numScores <= 3; numScores += 1)
     {
          cout << "Score " << numScores << ": ";
          cin >> score;
          sum += score;
     } //end for
     avg = sum / 3.0
     cout << "Average for student "
     << numStudents << ": " << avg << endl;
     numStudents += 1;
     cout << "Another student (Y/N)? ";
     cin >> anotherStud;
} while (anotherStud == 'Y' || anotherStud == 'y');
```
[or `} while (toupper(anotherStud) == 'Y');`]