NESTING LOOPS

In Chapter 7 you learned how to nest if structures. Loops can also be nested. In fact, loops within loops are very common. You must trace the steps of the program carefully to understand how ***nested loops*** behave. The program in Figure 8-12 provides output that will give you insight into the behavior of nested loops.

```
#include <iostream.h>

main()
{
 int i,j;
 cout << "BEGIN\n";
 for(i = 1; i <= 3; i++)
  {
   cout << " Outer loop: i = " << i << '\n';
   for(j = 1; j <= 4; j++)
    cout << "      Inner loop: j = " << j << '\n';
  }
 cout << "END\n";
 return 0;
}
```

FIGURE 8-12
Even though this program has little practical use, it illustrates what happens when loops are nested.

The important thing to realize is that the inner for loop (the one that uses `j`) will complete its count from 1 to 4 every time the outer for loop (the one that uses `i`) iterates. That is why in the output, for every loop the outer loop makes, the inner loop starts over (see Figure 8-13).

```
BEGIN
 Outer loop: i = 1
      Inner loop: j = 1
      Inner loop: j = 2
      Inner loop: j = 3
      Inner loop: j = 4
 Outer loop: i = 2
      Inner loop: j = 1
      Inner loop: j = 2
      Inner loop: j = 3
      Inner loop: j = 4
Outer loop: i = 3
      Inner loop: j = 1
      Inner loop: j = 2
      Inner loop: j = 3
      Inner loop: j = 4
END
```

FIGURE 8-13
The output of the program in Figure 8-12 illustrates the effect of the nested loops.

EXERCISE 8-8 NESTED LOOPS

1. Open *NESTLOOP.CPP*.
2. Compile and run the program. *Note:* If you know how to use your compiler's debugger, step through the program to trace the flow of logic. (See Appendix F for more information on debugging.)
3. Close the source file.

```
#include <iostream.h>

const char ERROR = '\0';

main()
{
 int num_reps;
 int i;          // counter used by for loop
 int democrats = 0, republicans = 0, independents = 0;
 char party;

 cout << "\nHow many U.S. representatives does your state have? ";
 cin  >> num_reps;  // ask user for number of representatives

 cout << "Enter the party affiliation for each representative.\n";
 cout << "Enter D for Democrat, R for Republican,\n";
 cout << "and I for independents or other parties.\n";
 for (i = 1; i <= num_reps; i++)
  {
   do
    {
     cout << "Party of representative #" << i << ": ";
     cin  >> party;
     switch(party)
      {
       case 'D':
       case 'd':          // if democrat,
        democrats++;      // increment democrats counter
        break;
       case 'R':
       case 'r':          // if republican,
        republicans++;    // increment republicans counter
        break;
       case 'I':
       case 'i':          // if independent or other,
        independents++;   // increment independents counter
        break;
       default:
        cout << "Invalid entry. Enter D, R, or I.\n";
        party = ERROR;
        break;
      } // end of switch structure
    } while (party == ERROR); // loop again if invalid choice is made
  } // end of for loop
 cout << "\nYour state is represented by " << democrats << " Democrats, "
      << republicans << " Republicans,\nand " << independents
      << " independents and members of other parties.\n\n";
 return 0;
} // end of program
```

FIGURE 8-14
This program has a do while loop nested within a for loop.