4.7 Nested loops

A ***nested loop*** is a loop that appears in the body of another loop. The nested loops are commonly referred to as the ***inner loop*** and ***outer loop***.

Nested loops have various uses. One use is to generate all combinations of some items. For example, the following program generates all two-letter .com Internet domain names.

Figure 4.7.1: Nested loops example: Two-letter domain name printing program.

```
#include <iostream>
using namespace std;

/* Output all two-letter .com Internet domain names */

int main() {
   char letter1;
   char letter2;

   cout << "Two-letter domain names:" << endl;

   letter1 = 'a';
   while (letter1 <= 'z') {
      letter2 = 'a';
      while (letter2 <= 'z') {
         cout << letter1 << letter2 << ".com" << endl;
         ++letter2;
      }
      ++letter1;
   }

   return 0;
}
```

```
Two-letter domain names:
aa.com
ab.com
ac.com
ad.com
ae.com
af.com
ag.com
ah.com
ai.com
aj.com
ak.com
al.com
am.com
an.com
ao.com
ap.com
aq.com
ar.com
as.com
at.com
au.com
av.com
aw.com
ax.com
ay.com
az.com
ba.com
bb.com
bc.com
bd.com
be.com

...

zw.com
zx.com
zy.com
zz.com
```

Feedback?

Note that the program makes use of ascending characters being encoded as ascending numbers, e.g., 'a' is 97, 'b' is 98, etc., so assigning 'a' to letter1 and then incrementing yields 'b'.

(Forget about buying a two-letter domain name: They are all taken, and each sells for several hundred thousand or millions of dollars. Source: dnjournal.com, 2012).

zyDE 4.7.1: Two character dotcom domain names.

Modify the program to include two-character .com names where the second char a letter or a number, as in a2.com. Hint: Add a second loop, following the `while (letter2 <= 'z')` loop, to handle numbers.

Load default template... Run

```
#include <iostream>
using namespace std;

/* Output all two-letter .com Internet dom

int main() {
   char letter1;
   char letter2;

   cout << "Two-letter domain names:" << en

   letter1 = 'a';
   while (letter1 <= 'z') {
      letter2 = 'a';
      while (letter2 <= 'z') {
         cout << letter1 << letter2 << ".co
         ++letter2;
      }
      ++letter1;
   }
```

Feedback?

Below is a nested loop example that graphically depicts an integer's magnitude by using asterisks, creating a "histogram." The inner loop is a for loop that handles the printing of the asterisks. The outer loop is a while loop that handles executing until a negative number is entered.

Figure 4.7.2: Nested loop example: Histogram.

```
#include <iostream>
using namespace std;

int main() {
   int numAsterisk;  // Number of asterisks to print
   int i;            // Loop counter

   numAsterisk = 0;

   while (numAsterisk >= 0) {
      cout << "Enter an integer (negative to quit): ";
      cin >> numAsterisk;

      if (numAsterisk >= 0) {
         cout << "Depicted graphically:" << endl;
         for (i = 1; i <= numAsterisk; ++i) {
            cout << "*";
         }
         cout << endl << endl;
      }
   }

   cout << "Goodbye." << endl;

   return 0;
}
```

```
Enter an integer (negative to quit): 9
Depicted graphically:
*********

Enter an integer (negative to quit): 23
Depicted graphically:
***********************

Enter an integer (negative to quit): 35
Depicted graphically:
***********************************

Enter an integer (negative to quit): -1
Goodbye.
```

Feedback?

PARTICIPATION ACTIVITY 4.7.1: Nested loops: Inner loop execution.

1) Given the following code, how many times will the inner loop body execute?

```
int row;
int col;

for(row = 0; row < 2; row = row + 1)
{
   for(col = 0; col < 3; col = col +
1) {
      // Inner loop body
   }
}
```

Check Show answer

2) Given the following code, how many times will the inner loop body

execute?

```
char letter1;
char letter2;

letter1 = 'a';
while (letter1 <= 'f') {
   letter2 = 'c';
   while (letter2 <= 'f') {
      // Inner loop body
      ++letter2;
   }
   ++letter1;
}
```

Check Show answer

Feedback?

PARTICIPATION ACTIVITY 4.7.2: Nested loops: What is the output.

1) What is output by the following code?

```
int row;
int col;

for(row = 2; row <= 3; row = row + 1)
{
   for(col = 0; col <= 1; col = col +
1) {
      cout << row << col << " ";
   }
}
```

Check Show answer

2) What is output by the following code?

```
char letter1;
char letter2;

letter1 = 'y';
while (letter1 <= 'z') {
   letter2 = 'a';
   while (letter2 <= 'c') {
      cout << letter1 << letter2 << "
";
      ++letter2;
   }
   ++letter1;
}
```

Check Show answer

Feedback?

CHALLENGE ACTIVITY 4.7.1: Nested loops: Indent text.

Print numbers 0, 1, 2, ..., userNum as shown, with each number indented by that number of spaces. For each printed line, print the leading spaces, then the number, and then a newline. Hint: Use i and j as loop variables (initialize i and j explicitly). Note: Avoid any other spaces like spaces after the printed number. Ex: userNum = 3 prints:

```
0
 1
  2
   3
```

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5     int userNum;
6     int i;
7     int j;
8
9     cin >> userNum;
10
11    /* Your solution goes here  */
12    for (int i = 0; i <=userNum; i++){
13       for(int j = 0; j < i; j++){
14          cout<< " ";
15       }
16       cout<< i<< endl;
17    }
18
19    return 0;
20 }
```

Run ✓ All tests passed

✓ Testing with userNum = 3

Your output

```
0
 1
```

```
  2
   3
```

✓ Testing with userNum = 6

Your output

```
0
 1
  2
   3
    4
     5
      6
```

✓ Testing with userNum = 0

Your output

```
0
```

✓ Testing with userNum = 20

Your output

```
0
 1
  2
   3
    4
     5
      6
       7
        8
         9
          10
           11
            12
             13
              14
               15
                16
                 17
                  18
                   19
                    20
```

Feedback?

CHALLENGE ACTIVITY 4.7.2: Nested loops: Print seats.

Given numRows and numColumns, print a list of all seats in a theater. Rows are numbered, columns lettered, as in 1A or 3E. Print a space after each seat, including after the last. Ex: numRows = 2 and numColumns = 3 prints:

```
1A 1B 1C 2A 2B 2C
```

```
2  using namespace std;
3
4  int main() {
5     int numRows;
6     int numColumns;
7     int currentRow;
8     int currentColumn;
9     char currentColumnLetter;
10
11    cin >> numRows;
12    cin >> numColumns;
13
14    /* Your solution goes here  */
15    for (currentRow = 1; currentRow <=numRows; currentRow++){
16       for(currentColumn = 1; currentColumn <= numColumns; currentColumn++){
17          currentColumnLetter = (char)(currentColumn+64);
18          cout<< currentRow<< currentColumnLetter <<" ";
19       }
20
21    }
22
23    cout << endl;
```

Run ✓ All tests passed

✓ Testing with 2 rows and 3 columns

Your output `1A 1B 1C 2A 2B 2C`

✓ Testing with 5 rows and 3 columns

Your output `1A 1B 1C 2A 2B 2C 3A 3B 3C 4A 4B 4C 5A 5B 5C`

✓ Testing with 5 rows and 0 columns

Your output

✓ Testing with 0 rows and 2 columns

Your output

Feedback?