

Lab 8 - CMPS 1043, Computer Science I

Lab 8 - For Loops

Lesson objectives:

Understand the control sequence of a **for** loop

Practice **for** loops

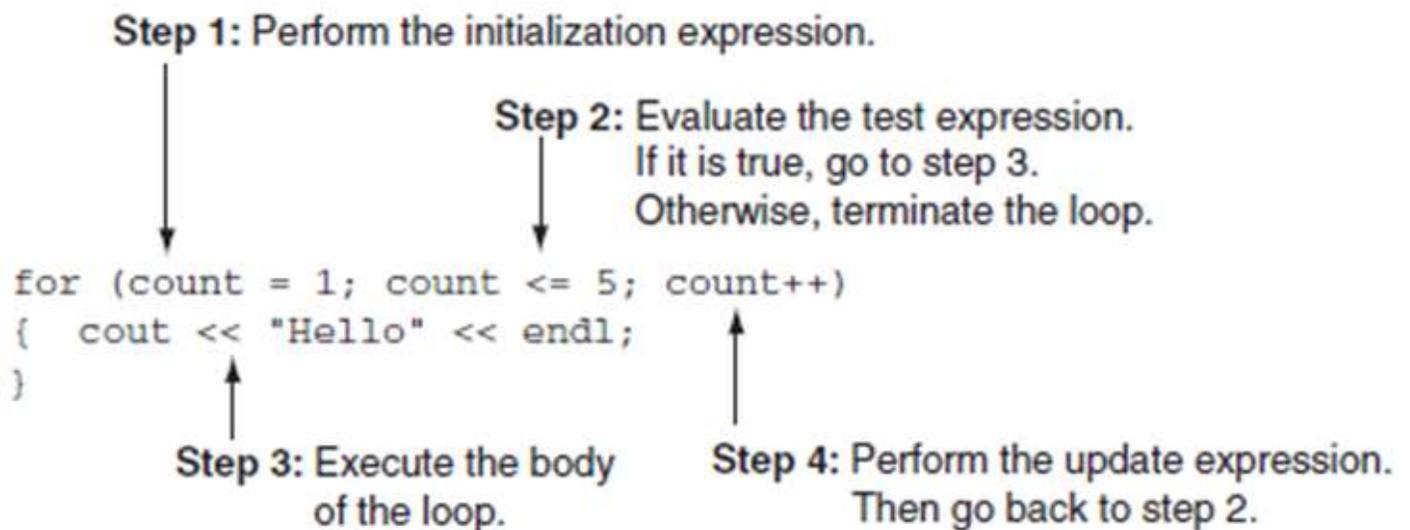
- a. using an increment
- b. using a decrement
- c. summation
- d. nested loops

As with other loop structures, the **for** loop contains an initialization, a test, and an update. Unlike other loop structures, these components are grouped together at the beginning of this pretest loop. **For** loops are often used when the number of iterations can be determined in advance.

Structure:

```
for (initialization; test; update)
{
    1 or more statements;
}
```

Example:



The code shown in the example will print hello for each value of count, 1-5. The value of count will increment to 6, but the loop will not execute at that point because the test expression will be false. The output of the example code is shown below.

```
Hello
Hello
Hello
Hello
Hello
```

The update in a **for** loop can be either an increment or a decrement. It is important to make sure that the test corresponds with the update.

Type in the following code:

```
#include<iostream>
#include<fstream>
#include<iomanip>
using namespace std;
int main()
{
    int count;
    for (count = 1; count <= 5; count++)
    {
        cout << count << '\n';
    }
    system("pause");
    return 0;
}
```

1. What is the output of the above code?
2. What happens if the update is changed to count-- ? Why?
3. Modify the code above so that the numbers are printed in reverse order.

The update can be incremented/decremented by more than one. For example, count+=2 will cause count to increment by two rather than one.

4. Modify the code to print all the even numbers from 2 to 100.
5. Modify the code to print: 10 20 30 40 50 60 70 80 90 100

An accumulator can be used with a **for** loop to keep a running total of a value. For example, the code below will add all the numbers from 1 to 100.

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

int main()
{
    int count, sum;
    sum = 0;
    for (count = 1; count <= 100; count++)
    {
        sum += count;
    }
    cout << "The sum of the numbers from 1-100 is " << sum << '\n';
    system("pause");
    return 0;
}
```

6. Type in and run the code above. What is the output?

The term *nested loops* means that one loop is completely inside another. When nested loops are used, the inner loop goes through all its repetitions for each iteration of the outer loop. The total number of repetitions for an inner loop is the product of the number of repetitions of the two (or more) loops. The code below illustrates a nested **for** loop which prints a 5 X 4 table where each table entry represents the appropriate row number.

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

int main()
{
    int r, c;
    for (r = 1; r <= 5; r++)
    {
        for (c = 1; c <= 4; c++)
        {
            cout << setw(3) << r;
        }
        cout << '\n';
    }
    system("pause");
    return 0;
}
```

7. Type in and run the code above. Modify the code so that it prints the column number rather than the row number.

Assignment

Modify the last program to include the following.

1. Add header of comments containing required information (as specified in the previous labs). Add at least 3 comments in the body of the program.
2. Print a 10 X 10 table in which each entry in the table is the sum of the row and column number.
3. Include an accumulator that will calculate the sum of all the table entries. Print the sum outside of the nested loop along with an appropriate output statement.
4. When the program is working correctly, change all cout statements to statements that will print to a file, as you learned in an earlier lab. Print out both the program code and the output file to turn in. SAVE your work before leaving the lab.