

CIS 22A – Lecture 11

Manish Goel

The `for` Loop

- Useful for counter-controlled loop
- General Format:

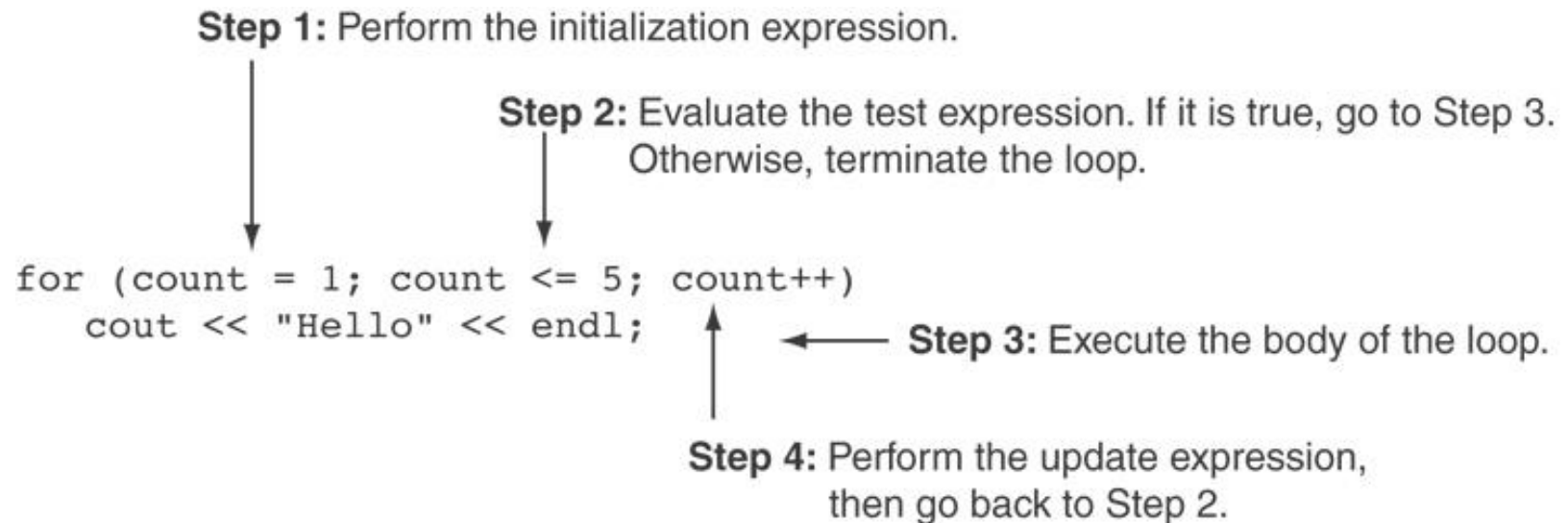
```
for(initialization; test; update)  
    statement; // or block in { }
```

- 1) Perform *initialization*
 - 2) Evaluate *test* expression
 - If `true`, execute *statement*
 - If `false`, terminate loop execution
 - 3) Execute *update*, then re-evaluate *test* expression
- No semicolon after the `update` expression or after the `)`

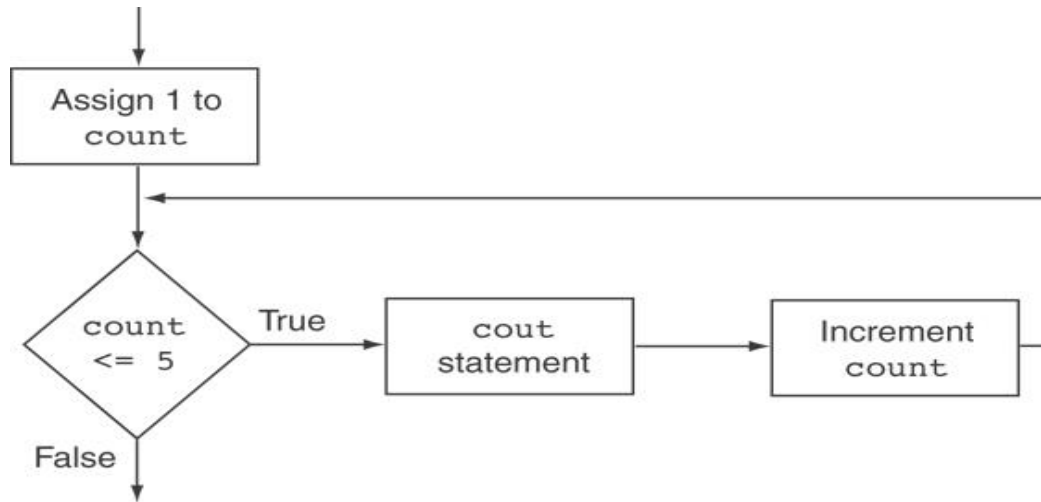
for Loop - Example

```
int count;
```

```
for (count = 1; count <= 5; count++)  
    cout << "Hello" << endl;
```



for Loop – things to consider



- In any situation that clearly requires
 - an initialization
 - a false condition to stop the loop
 - an update to occur at the end of each iteration
- The `for` Loop is a Pretest Loop

A for Loop in Program 5-9

Program 5-9

```
1 // This program displays the numbers 1 through 10 and
2 // their squares.
3 #include <iostream>
4 using namespace std;
5
6 int main()
7 {
8     const int MIN_NUMBER = 1,    // Starting value
9           MAX_NUMBER = 10;    // Ending value
10    int num;
11
12    cout << "Number Number Squared\n";
13    cout << "-----\n";
14
15    for (num = MIN_NUMBER; num <= MAX_NUMBER; num++)
16        cout << num << "\t\t" << (num * num) << endl;
17
18    return 0;
19 }
```

Program Output

Number	Number Squared
1	1
2	4
3	9
4	16
5	25
6	36
7	49
8	64
9	81
10	100

for Loop - Modifications

- You can have multiple statements in the *initialization* expression. You can also have multiple statements in the *test* expression. Separate the statements with a comma:

```
int x, y;
for (x=1, y=1; x <= 5; x++, y++)
{
    cout << x << " plus " << y
        << " equals " << (x+y)
        << endl;
}
```

Initialization Expression Test Expression

initialization expression can be omitted if already done:

```
int sum = 0, num = 1;
for (; num <= 10; num++)
    sum += num;
```

Variables can be declared in the *initialization* expression - the scope of the variable `num` is the `for` loop.

```
int sum = 0;
for (int num = 0; num <= 10;
    num++)
    sum += num;
```

Nested Loops

- A nested loop is a loop inside the body of another loop
- Inner (inside), outer (outside) loops:

```
for (row=1; row<=3; row++) //outer
    for (col=1; col<=3; col++)//inner
        cout << row * col << endl;
```

- Inner loop goes through all repetitions for each repetition of outer loop
- Inner loop repetitions complete sooner than outer loop
- Total number of repetitions for inner loop is product of number of repetitions of the two loops.

Nested for Loop in Program 5-14

```
26 // Determine each student's average score.
27 for (int student = 1; student <= numStudents; student++)
28 {
29     total = 0; // Initialize the accumulator.
30     for (int test = 1; test <= numTests; test++)
31     {
32         double score;
33         cout << "Enter score " << test << " for ";
34         cout << "student " << student << ": ";
35         cin >> score;
36         total += score;
37     }
38     average = total / numTests;
39     cout << "The average score for student " << student;
40     cout << " is " << average << ".\n\n";
41 }
```

Inner Loop

Outer Loop