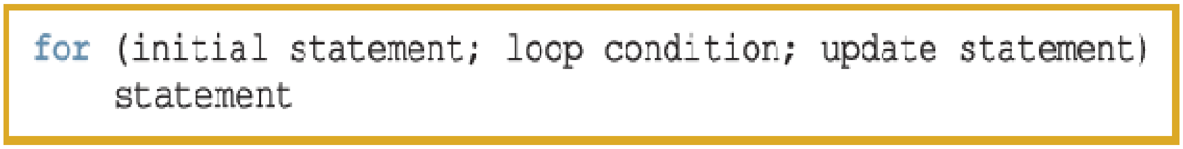
Programming Language (630203)
Fall 2010/2011 – Lecture Notes # 10

Repetition Control Structures-III (for structure)

- The general form of the **for** statement is:



```
for (initial statement; loop condition; update statement)
    statement
```

- The initial statement, loop condition, and update statement are called for loop control statements
- Initial statement usually initializes a variable (called the for **loop control**, or for **indexed, variable**).
- In C++, **for** is a reserved word

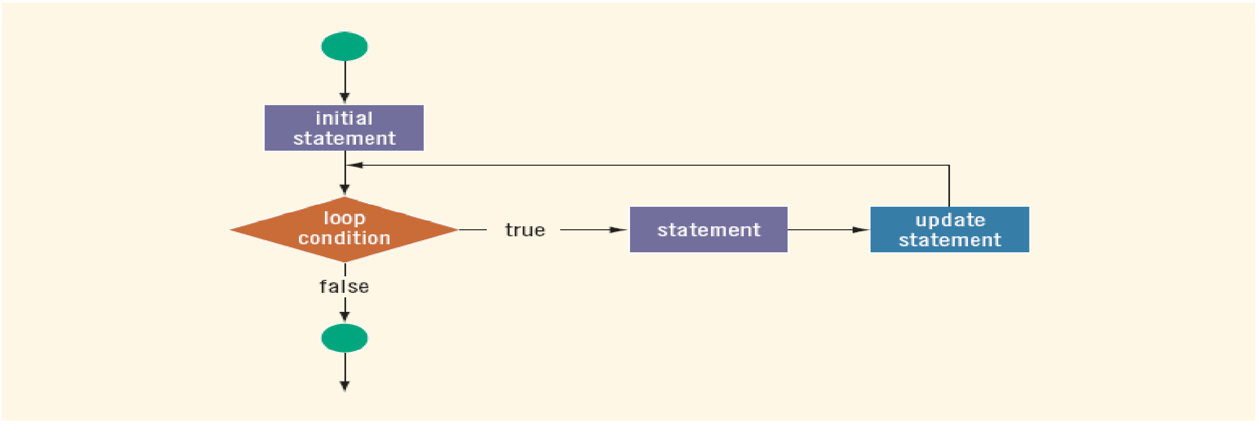
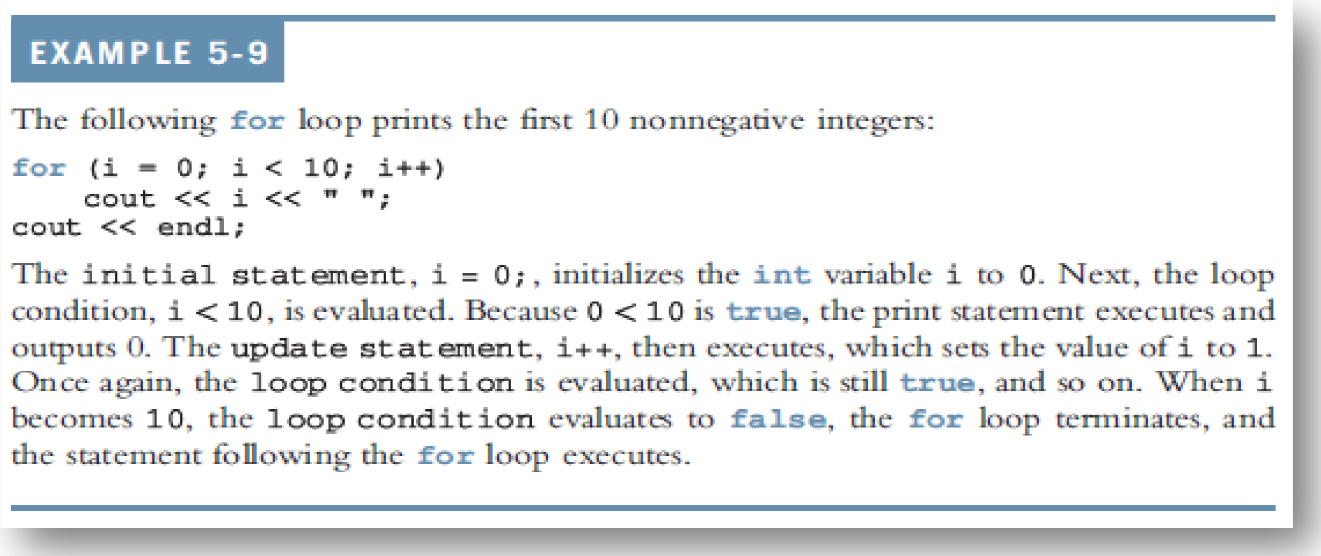


FIGURE 5-2 for loop



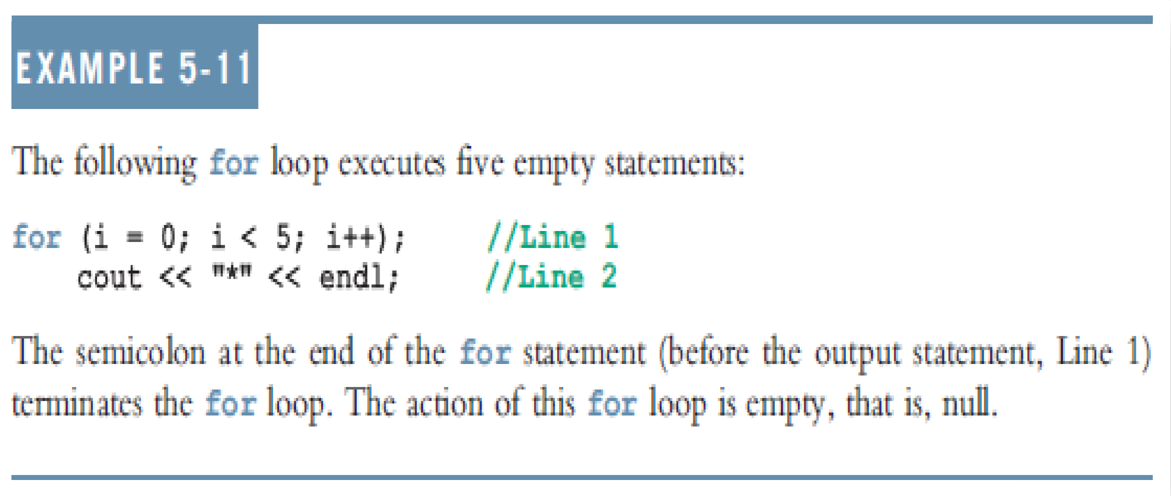
EXAMPLE 5-9

The following for loop prints the first 10 nonnegative integers:

```
for (i = 0; i < 10; i++)
    cout << i << " ";
cout << endl;
```

The initial statement, i = 0;, initializes the int variable i to 0. Next, the loop condition, i < 10, is evaluated. Because 0 < 10 is true, the print statement executes and outputs 0. The update statement, i++, then executes, which sets the value of i to 1. Once again, the loop condition is evaluated, which is still true, and so on. When i becomes 10, the loop condition evaluates to false, the for loop terminates, and the statement following the for loop executes.

- C++ allows you to use fractional values for loop control variables of the double type
    - Results may differ
- The following is a semantic error:



EXAMPLE 5-11

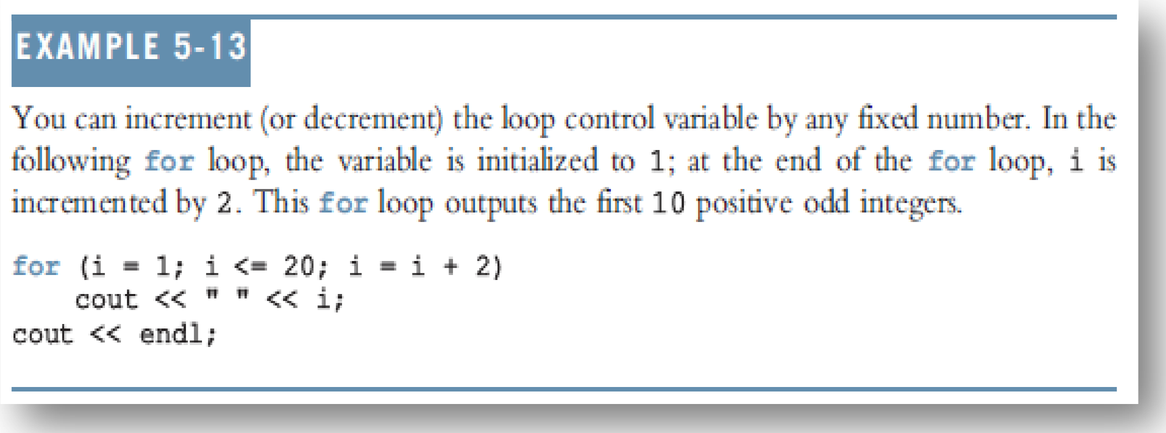
The following for loop executes five empty statements:

```
for (i = 0; i < 5; i++);      //Line 1
    cout << "*" << endl;      //Line 2
```

The semicolon at the end of the for statement (before the output statement, Line 1) terminates the for loop. The action of this for loop is empty, that is, null.

- The following is a legal for loop:

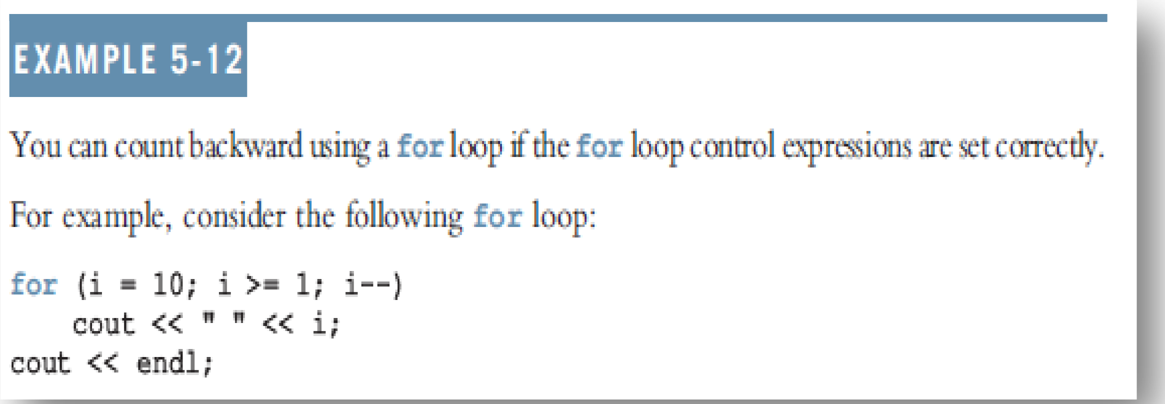
```
for (;;)
 cout << "Hello" << endl;
```



EXAMPLE 5-13

You can increment (or decrement) the loop control variable by any fixed number. In the following for loop, the variable is initialized to 1; at the end of the for loop, i is incremented by 2. This for loop outputs the first 10 positive odd integers.

```
for (i = 1; i <= 20; i = i + 2)
    cout << " " << i;
cout << endl;
```



EXAMPLE 5-12

You can count backward using a for loop if the for loop control expressions are set correctly.

For example, consider the following for loop:

```
for (i = 10; i >= 1; i--)
    cout << " " << i;
cout << endl;
```

- The for statement (also called for loop) is similar to the while statement The general form of the **for** statement is:

```
                  for (expression1; expression2; expression3)
                            statement;
```

- First expression1 is evaluated. Each time round the loop, expression2 is evaluated. If the outcome is nonzero then statement is executed and expression3 is evaluated. Otherwise, the loop is terminated. The general for loop is equivalent to the following while loop:

```
                  expression1;
                  while (expression2) {
                       statement;
                  expression3;
                  }
```

Example 1

```
//Program to determine the sum of the first n positive numbers.
#include <iostream>
using namespace std;
int main()
{
    int counter;    //loop control variable
    int sum;        //variable to store the sum of numbers
    int N;          //variable to store the number of
                    //first positive integers to be added
    cout << "Line 1: Enter the number of positive "
         << "integers to be added: ";                //Line 1
    cin >> N;                                        //Line 2
    sum = 0;                                         //Line 3
    cout << endl;                                    //Line 4
    for (counter = 1; counter <= N; counter++)       //Line 5
        sum = sum + counter;                         //Line 6
    cout << "Line 7: The sum of the first " << N
         << " positive integers is " << sum
         << endl;                                    //Line 7
    return 0;
}
```

Example 2

```
//*******************************************************
// Program: Counts zeros, odds, and evens
// This program counts the number of odd and even numbers.
// The program also counts the number of zeros.
//********************************************************
#include <iostream>
#include <iomanip>
using namespace std;
```

```
const int N = 20;
int main ()
{
        //Declare variables
    int counter;    //loop control variable
    int number;     //variable to store the new number
    int zeros = 0;                                  //Step 1
    int odds = 0;                                   //Step 1
    int evens = 0;                                  //Step 1
    cout << "Please enter " << N << " integers, "
         << "positive, negative, or zeros."
         << endl;                                   //Step 2
    cout << "The numbers you entered are:" << endl;
    for (counter = 1; counter <= N; counter++)      //Step 3
    {
        cin >> number;                              //Step 3a
        cout << number << " ";                      //Step 3b
            //Step 3c
        switch (number % 2)
        {
        case 0:
            evens++;
            if (number == 0)
                zeros++;
            break;
        case 1:
        case -1:
            odds++;
        } //end switch
    } //end for loop
    cout << endl;
                    //Step 4
    cout << "There are " << evens << " evens, "
         << "which includes " << zeros << " zeros."
         << endl;
    cout << "The number of odd numbers is: " << odds
         << endl;
    return 0;
}
```