Mr. Giansante

C++ Programming

Loops

August 2018

For Loops

Source: Learn C++ Programming Language, TutorialsPoint, 2014

C++, like most computer languages, provides several looping structures.

For Loops

A for loop is a repetition control structure that allows you to efficiently write a loop that needs to execute a specific number of times.

```
for ( init; condition; increment )
{
     statement(s);
}
```

Example:

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

int main ()
{
     // for loop execution
     for( int a = 10; a < 20; a = a + 1 )
     {
          cout << "value of a: " << a << "\n";
     }

     return 0;
}
```

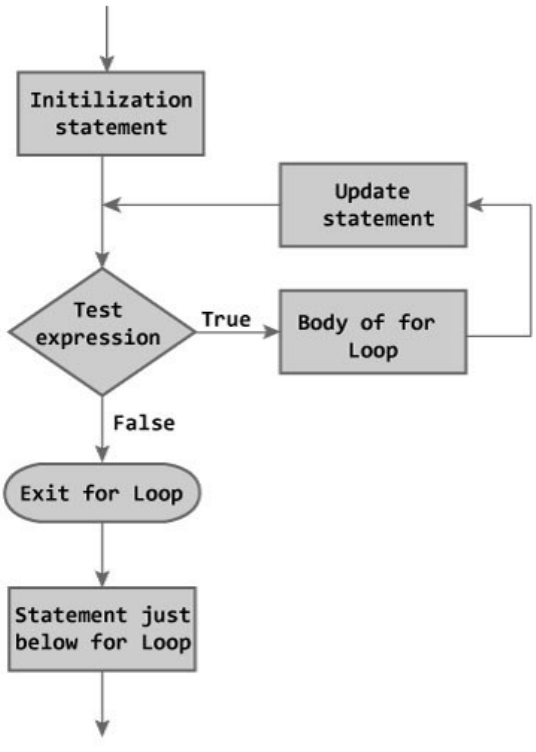

Figure: Flowchart of for Loop

Exiting the Loop Before it is Done

The `break` statement terminates the loop or switch statement and transfers execution to the statement immediately following the loop or switch.

The Infinite Loop

A loop becomes infinite loop if a condition never becomes false. The for loop is traditionally used for this purpose. Since none of the three expressions that form the 'for' loop are required, you can make an endless loop by leaving the conditional expression empty.

"Nesting" Loops

You can nest loops by placing one loop within another. Give each loop a unique variable name as its counter.

```
for (int i = 1; i < 11; i = i + 1)
{
     for (int j = 1; j < 6; j = j + 2)
     {
          for (int k = 100; k > 0; k = k - 1)
          {
               ...
          }
     }
}
```

Example: The following program uses a nested for loop to find the prime numbers from 2 to 100:

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

int main ()
{
     int i, j;
     bool prime = false;

     for(i=2; i<100; i++)
     {
          // Assume the number is prime
          prime = true;

          for(j=2; j <= (i/j); j++)
          {
               // If factor found, set prime=false
               if(i%j == 0)
               {
                    prime = false;
               }
          }
          if(prime == true)
          {
              cout << i << " is a prime number.\n";
          }
     }
     return 0;
}
```

While and Do - While Loops

While Loop

A while loop statement repeatedly executes a target statement as long as a given condition is true.

```
while(condition)
{
     statement(s);
}
```

Example: Counting from 1 to 10

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

int main ()
{
     int x = 0;

     while(x <= 10)
     {
          cout << x << "\n";
     }

     return 0;
}
```

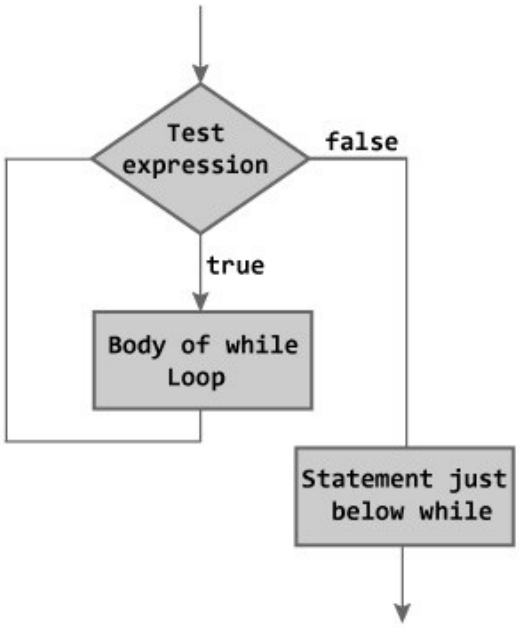

Figure: Flowchart of while Loop

Do - While Loop

Unlike for and while loops, which test the loop condition at the top of the loop, the do...while loop checks its condition at the bottom of the loop.

A do...while loop is similar to a while loop, except that a do...while loop is guaranteed to execute at least one time.

```
do
{
     statement(s);
}while( condition );
```

Example: Counting from 1 to 10

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

int main ()
{
     int x = 1;

     do
     {
          cout << x << "\n";
     }while(x <= 10)

     return 0;
}
```

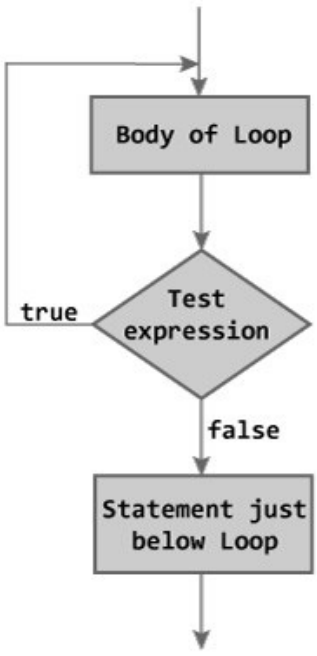

Figure: Flowchart of do...while Loop