FAST National University of Computer and Emerging Sciences Peshawar

OOP Lab # 3.2

DEPARTMENT OF COMPUTER SCIENCE

C++ Programming (Loops)

Computer Instructor: Muhammad Abdullah Orakzai

Prepared By: Muhammad Abdullah Orakzai (Lab Instructor CS)

Contents

Control Structure/Loop

- In computer programming, a **loop** is a sequence of instructions that is continually repeated until a certain condition is reached.
- A **loop** statement allows us to execute a statement or group of statements multiple times.
- Loops are used in programming to repeat a specific block until some end condition is met.

Control Structure/Loop…

There are four type of loops in C++ programming:

1) for loop
2) while loop
3) do while loop
4) for-each loop (Enhanced For Loop)

Control Structure/Loop…

Pretested Loop: Loop in which condition is checked first.

e.g. for loop and while loop

Post tested Loop: Loop in which condition is checked at the end.

e.g. do while loop

Determined loop/Definite loop: Loop for fixed repetition

e.g. for loop

Undetermined loop/Indefinite loop: loop not for fixed repetition

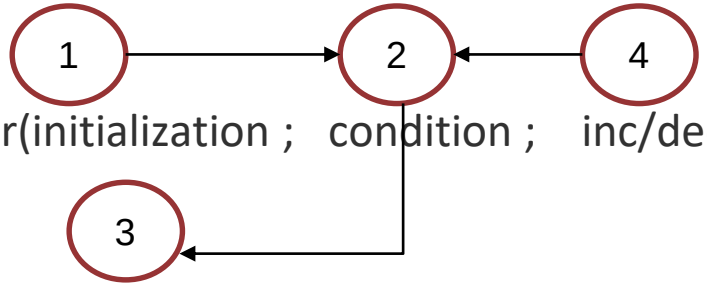
E.g. while loop and do while loop

1) for loop

- for loop is used to a statement or group of statement for a fixed number of time.
- If the number of iteration is fixed then it is recommended to use for loop.
- 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.

1) for loop…

Syntax:




```
for(initialization ;   condition ;    inc/dec)
{
// Statement(s)

}
```

1) for loop…

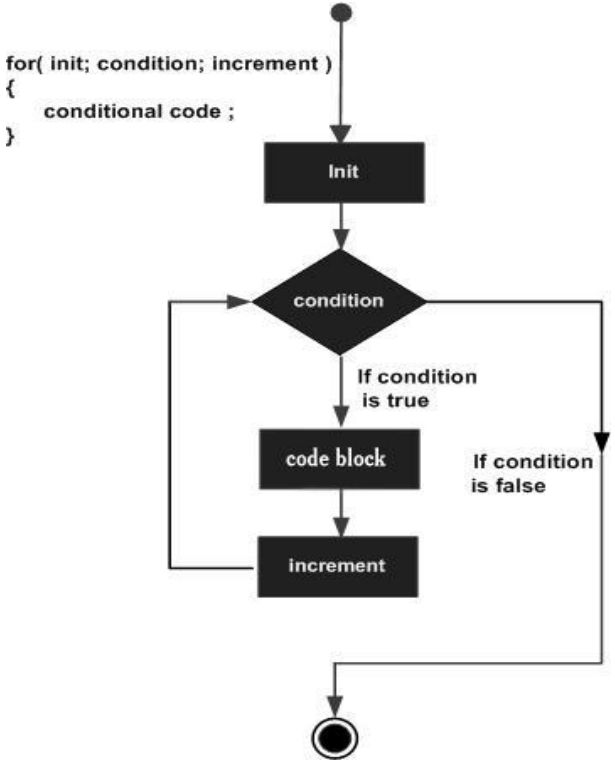
Order of Steps in for loop:

1) 1st initialization is performed.
2) Secondly condition is checked
3) In 3rd step statement is executed means control goes to body of the loop.
4) In 4th step incrementation or decrementation is performed.
5) Again condition is checked an so on.

Note: In loops variable is called counter variable.

i= i+1; OR i+=1;

1) for loop…


for(init; condition; increment)
{
conditional code ;
}
Init
condition
If condition
is true
code block
If condition
is false
increment

1) for loop…

```
#include <iostream>
using namespace std;
int main() {

for (int i = 0 ; i < 5 ; i++)
{
    cout << i << "\n";
}


  return 0;
}
```

Output:

```
0
1
2
3
4
```

1) for loop…

```
#include <iostream>
using namespace std;
int main ()
{
   // for loop execution

   for( int a = 10 ; a < 20; a++ )
   {
      cout << "value of a: " << a << endl ;
   }

return 0;
}
```

```
value of a: 10
value of a: 11
value of a: 12
value of a: 13
value of a: 14
value of a: 15
value of a: 16
value of a: 17
value of a: 18
value of a: 19
```

1) for loop…

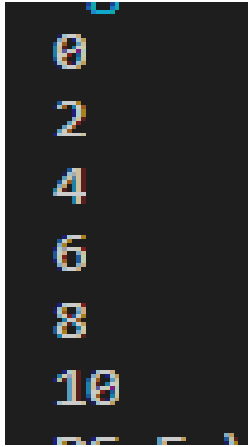
```
#include <iostream>
using namespace std;
int main ()
{
for( int a = 0; a < =10; a++ )
    {
     cout << "value of a: " << a << endl;
    }
return 0;
}
```

Output:

value of a: 0
value of a: 1
value of a: 2
value of a: 3
value of a: 4
value of a: 5
value of a: 6
value of a: 7
value of a: 8
value of a: 9
Value of a: 10

1) for loop…

```
#include <iostream>
using namespace std;
int main() {
  for (int i = 0 ; i <= 10; i = i + 2)
{
    cout << i << "\n";
}
  return 0;
}
```



infinitive for loop

If you use two semicolons (; ;) in the for loop it will be infinitive for loop.

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

int main () {
 for (;;)
 {
    cout<<“infinitive for loop";
 }

   return 0;
}
```

for loop Tasks

1. Write a C++ program which display first 10 number using for loop.
2. Write a C++ program which display even and odd number using for loop.
3. Take a number from user and make a table of that number using for loop.
4. Take a number from user and find factorial of that number using for loop.

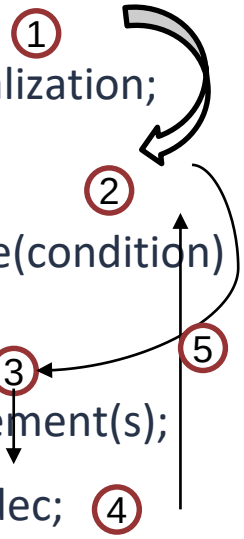
2) while loop

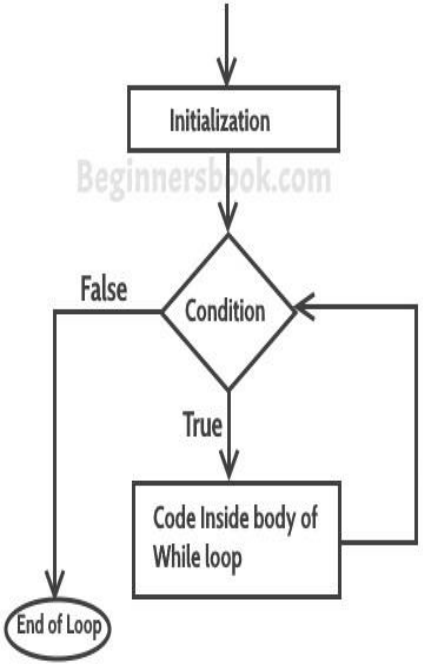
- Is used when number of iteration is not fixed.
- A **while** loop statement repeatedly executes a target statement as long as a given condition is true.

2) while loop

Syntax

```
initialization;
while(condition)
{
statement(s);
inc/dec;
}
```





2) while loop

- Here, **statement(s)** may be a single statement or a block of statements. The **condition** may be any expression, and true is any non-zero value. The loop iterates while the condition is true.
- When the condition becomes false, program control passes to the line immediately following the loop.

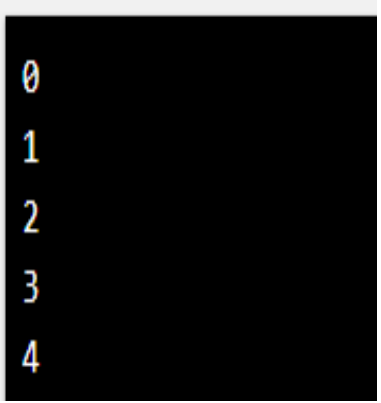
2) while loop…

```
#include <iostream>
using namespace std;
int main() {
  int i = 0;

  while (i < 5) {

   cout << i << "\n";
    i++;
  }

  return 0;
}
```

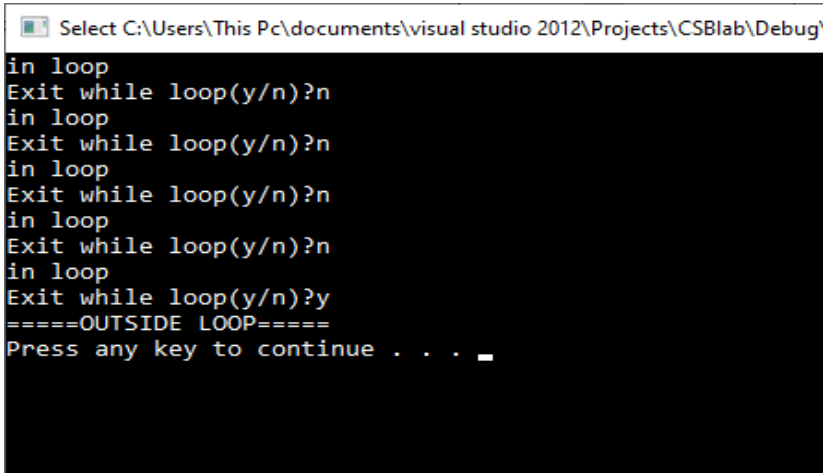


Note: Do not forget to increase the variable used in the condition, otherwise the loop will never end!

2) while loop…

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

int main () {
   // Local variable declaration:
    char c = 'n';
   // while loop execution
   while( c !='y' )
   {
      cout << "in loop"<<endl;
      cout<<"Exit while loop(y/n)?";
      cin>>c;
   }
   cout<<"=====OUTSIDE LOOP====="<<endl;
   return 0;
}
```



Sentinel Condition: Truthfulness or falseness depends upon user input.

infinitive while loop

If you pass true or true value in the while loop, it will be infinitive while loop.

Syntax:

```
while(true)
{
   Statement(s);
 }
```

infinitive while loop…

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

int main () {

 while(true)
 {
    cout<<"infinitive while loop";
 }

   return 0;
}
```

while loop Tasks

1. Write a C++ program which display first 10 number using while loop.
2. Write a C++ program which display even and odd number using while loop.
3. Take a number from user and make a table of that number using while loop.
4. Take a number from user and find factorial of that number using while loop.
5. Make a calculator using if-else-if else statement which perform the addition, subtraction, multiplication, division and remainder operations. Take values and operator from user on runtime. Use while loop for user choice. Means after performing one operation program will ask from user “do you want to do another calculation(yes/no)? ”. If user press then user will enter number 1, number 2 and operator for calculation and if user press no then terminate the loop.

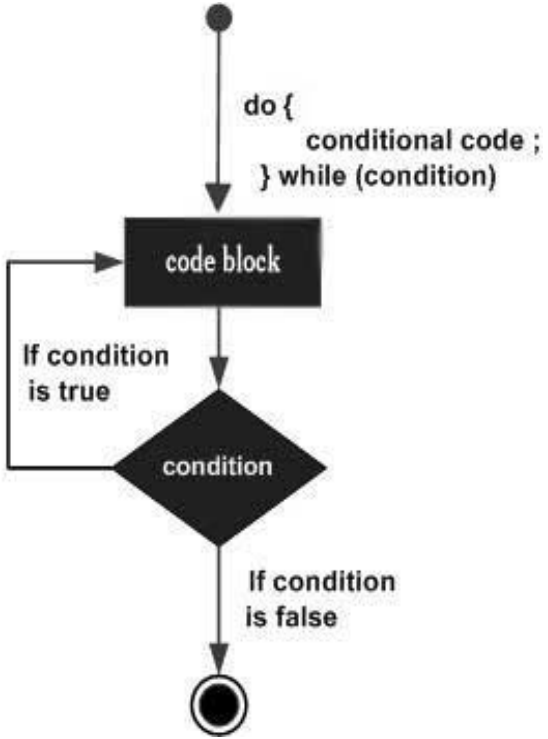
3) do while loop

- An indefinite loop. Best used when the number of iteration is unknown.
- Used when you will execute the loop at least once.
- 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.

3) do while loop…

Syntax

```
initialization;
do
{
statement(s);
inc/dec;
}
while(condition) ;
```



3) do while loop…

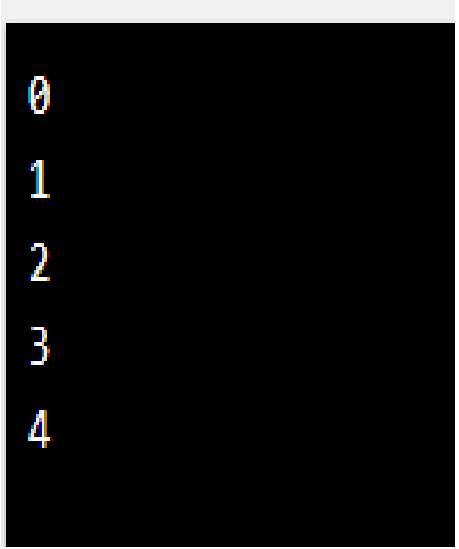
- Notice that the conditional expression appears at the end of the loop, so the statement(s) in the loop execute once before the condition is tested.
- If the condition is true, the flow of control jumps back up to do, and the statement(s) in the loop execute again. This process repeats until the given condition becomes false.

3) do while loop…

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

int main() {
  int i = 0;
  do {
    cout << i << "\n";
    i++;
  }
  while (i < 5);

  return 0;
}
```

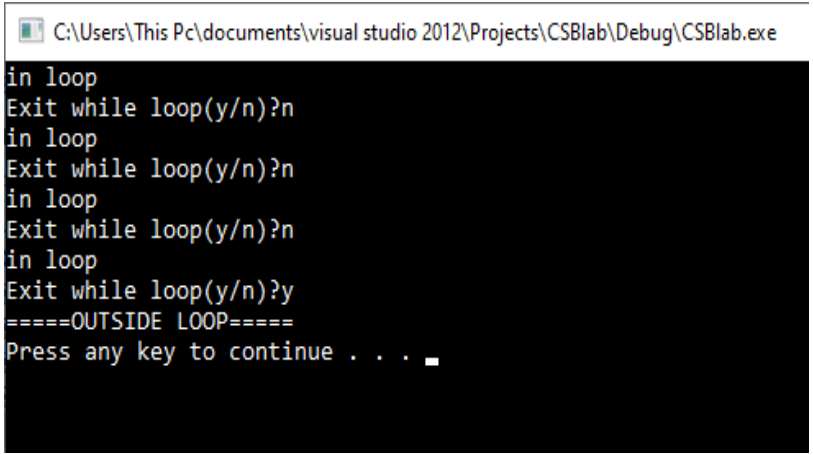


Do not forget to increase the variable used in the condition, otherwise the loop will never end!

3) do while loop…

```
#include <iostream>
using namespace std;
int main ()
{
   // Local variable declaration:
      char c ;
   // do-while loop execution
      do
      {
       cout << "in loop"<<endl;
       cout<<"Exit while loop(y/n)?";
       cin>>c;
   }
while( c !='y' );

   cout<<"=====OUTSIDE LOOP====="<<endl;
   return 0;
}
```



infinitive do while loop

If you pass true or true value in the do while loop, it will be infinitive do while loop.

Syntax:

```
do
{
Statement(s);
}
while(true);
```

infinitive do while loop…

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

int main () {
 while(true)
 {
    cout<<"infinitive do while loop";
 }

   return 0;
}
```

do while loop Tasks

1. Write a C++ program which display first 10 number using do while loop.
2. Write a C++ program which display even and odd number using while loop.
3. Take a number from user and make a table of that number using do while loop.
4. Take a number from user and find factorial of that number using do while loop.
5. Make a calculator using if-else-if else statement which perform the addition, subtraction, multiplication, division and remainder operations. Take values and operator from user on runtime. Use do while loop for user choice. Means after performing one operation program will ask from user “do you want to do another calculation(yes/no)? ”. If user press then user will enter number 1, number 2 and operator for calculation and if user press no then terminate the loop.

Enhanced for loop (for-each loop)

- Works with array.
- Is used for traversing in array.
- It is easy to use than simple for loop because here we do not initialization of counter variable, condition and incrementation or decrementation of counter variable.

Syntax

for (datatype variable-name : arrayname)

{ statement(s); }

Data type must be same as that of array data type

Enhanced for loop (for-each loop)...

```
#include<iostream>
using namespace std;
int main()
{
    int arr[]={1,2,3,4,5};   //array initialization

    cout<<"The elements are: ";
    for(int i : arr)
    {
        cout<<i<<" ";
    }
    return 0;
}
```

```
The elements are: 1 2 3 4 5
```

Break Statement

The break statement terminates the execution of the loop when it is used inside the body of the loop.

Syntax: break;

The break statement can also be used to jump out of a **loop**.

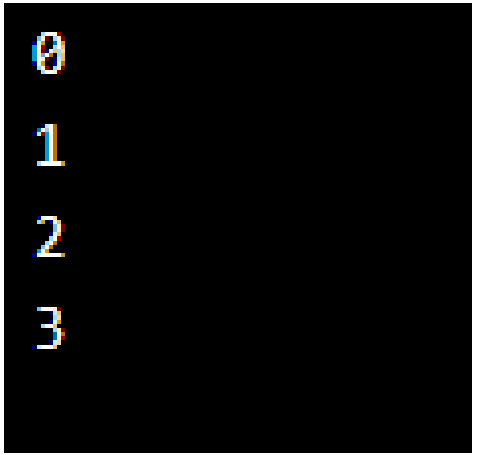
Break Statement Example

- This example jumps out of the loop when `i` is equal to 4:

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

int main() {
for (int i = 0; i <10; i++)
{
    if (i == 4) {
      break;
    }
    cout << i << "\n";
}


  return 0;
}
```



Continue Statement

- The continue statement shifts the control back to the beginning of the loop.
- It is used inside the body of the loop.
- It is used to continue loop.
- It continues the current flow of the program and skips the remaining code at specified condition.

Syntax: continue;

Continue Statement Example: Use of continue in for loop

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

int main() {
  for (int i = 0; i < 10; i++)
{
    if (i == 4) {
      continue;
    }
    cout << i << "\n";
  }
  return 0;
}
```

```
0
1
2
3
5
6
7
8
9
```

Nested Loop

- Loop within the body of another loop is called nested loop.

Nested for loop

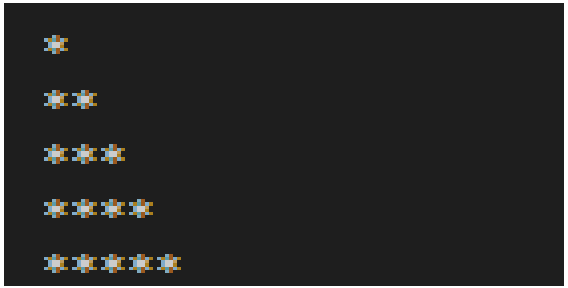
Program 1: (Triangular loop which make triangle using for nested loop with help of astricts (*)).

```
*
**
***
****
*****
```

Nested for loop…

```
#include <iostream>
using namespace std;
int main () {
 for (int i=1 ; i<=5 ; i++)
 {
    for (int j=0 ; j<i ; j++)
    {
        cout<<"*";
    }
 cout<<endl;
 }

   return 0;
}
```



Nested for loop…

Program 2: (Triangular loop which make triangle using for nested loop with help of numbers).

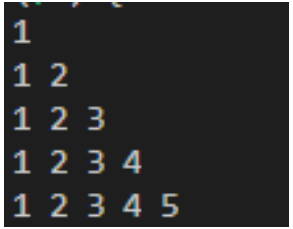
```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

Nested for loop…

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

int main () {
 for (int i=1 ; i<=5 ; i++)
 {
    for (int j=1 ; j<=i ; j++)
    {
        cout<<j;
        cout<<" ";
    }
 cout<<endl;
 }

   return 0;
}
```



Nested for loop tasks

1. Write a C++ program that will display * in the following pattern.

```
*****
****
***
**
*
```

2. Write a C++ program that will display numbers in the following pattern.

```
1 2 3 4 5
1 2 3 4
1 2 3
1 2
1
```

Summary

1	for loop Execute a sequence of statements multiple times and abbreviates the code that manages the loop variable.
2	while loop Repeats a statement or group of statements while a given condition is true. It tests the condition before executing the loop body.
3	do...while loop Like a ‘while’ statement, except that it tests the condition at the end of the loop body.
4	nested loops You can use one or more loop inside any another ‘while’, ‘for’ or ‘do..while’ loop.

References

- https://beginnersbook.com/2017/08/cpp-data-types/
- https://www.geeksforgeeks.org/c-data-types/
- http://www.cplusplus.com/doc/tutorial/basic_io/
- https://www.geeksforgeeks.org/basic-input-output-c/
- https://www.w3schools.com/cpp/default.asp
- https://www.javatpoint.com/cpp-tutorial

THANK YOU