Programming in c++
Lecture_4
Loops

Dr. Sara Mohamed

Simple Loops

- When an action must be repeated, a loop is used
- C++ includes several ways to create loops
- We start with the *while*-loop

◆ Syntax:

```cpp
while (condition) {
    // code to be executed
}
```

- **condition** → is checked **before** each iteration.
- If it's **true**, the loop body executes.
- If it's **false**, the loop stops.

The Power of Loops

- Computers are dumb machines, that only do what they are told.
- Their power lies in that they follow instructions very quickly, and don’t mind repeating the same instruction millions of times per second. Hence the power of the loop.
- The programmer's job is then to tell the computer how to loop, when to stop looping, and what to do each pass through the loop

While Loop Operation

The `while` loop loops through a block of code as long as a specified condition is `true`:

Syntax

```
while (condition) {
  // code block to be executed
}
```

The simplest loops to understand are while loops, which continue looping as long as some condition remains true.

Syntax of the *while* Statement

A Loop Body with Several Statements:

```
while (Boolean_Expression )
{
      Statement_1
      Statement_2
        . . .
      Statement_Last
}
```

Do NOT put a semicolon here.

body

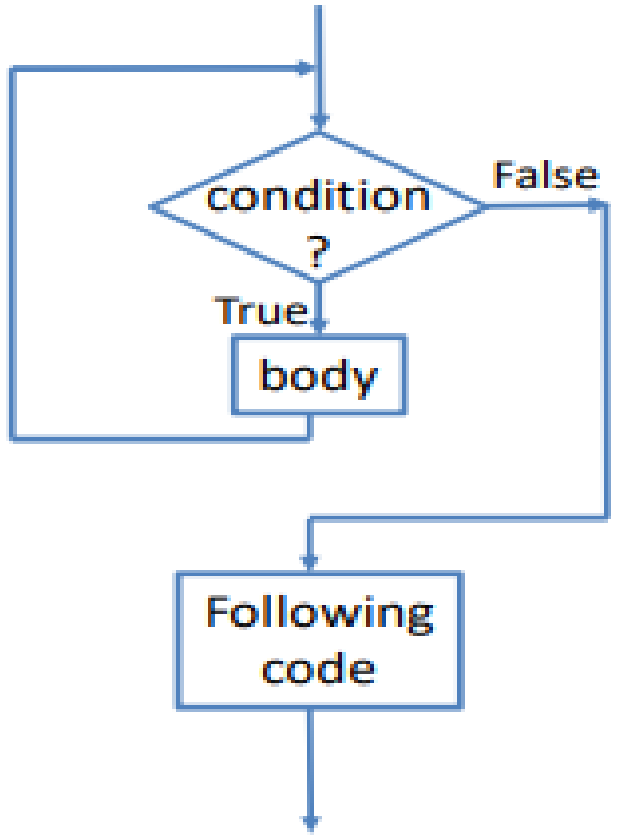
A Loop Body with a Single Statement:

```
while (Boolean_Expression )
      Statement
```

body

```
while( condition ) {
        // code in body of loop
}
```

- If braces are omitted, then a single statement comprises the body of the loop.



While Loop Operation

- First, the boolean expression is evaluated
  - If false, the program skips to the line following the while loop
  - If true, the body of the loop is executed
    - During execution, some item from the boolean expression is changed
  - After executing the loop body, the boolean expression is checked again repeating the process until the expression becomes false
- A while loop might not execute at all if the boolean expression is false on the first check

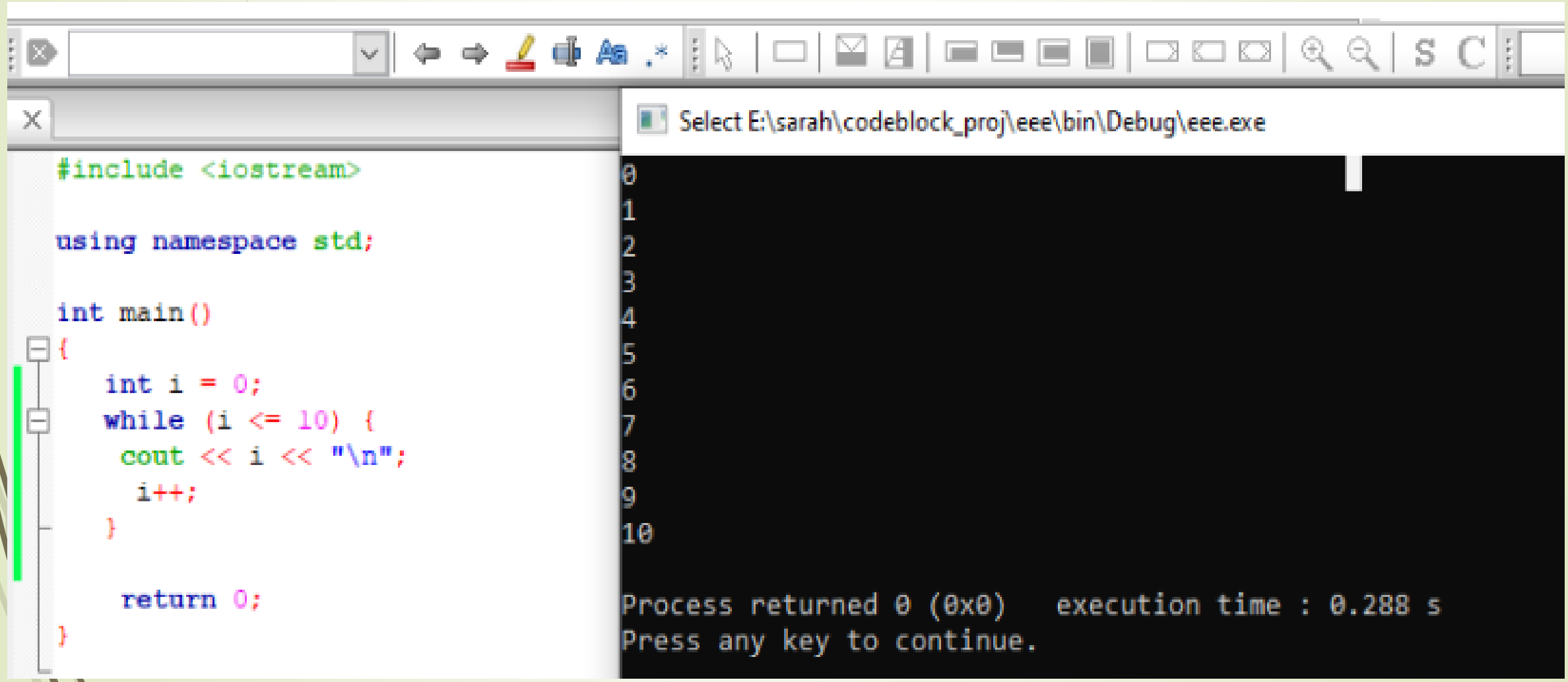
Write a c++ program print Hello 10 times

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

int main() {
    int count = 1;  // initialize counter

    while (count <= 10) {
        cout << "Hello" << endl;
        count++;  // increase counter to avoid infinite loop
    }

    return 0;
}
```

Output:

```
nginx


Hello
Hello
Hello
Hello
Hello
Hello
Hello
Hello
Hello
Hello
Hello
```

Write a c++ program to print number from 0 to 10?



```cpp
#include <iostream>

using namespace std;

int main()
{
    int i = 0;
    while (i <= 10) {
     cout << i << "\n";
      i++;
    }

     return 0;
}
```

Select E:\sarah\codeblock_proj\eee\bin\Debug\eee.exe

```
0
1
2
3
4
5
6
7
8
9
10

Process returned 0 (0x0)   execution time : 0.288 s
Press any key to continue.
```

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

int main() {
    int i = 1;  // start from 1

    while (i <= 50) {
        cout << i << endl;   // print the number
        i += 2;              // move to the next odd number
    }

    return 0;
}
```

Output:

python-repl

```
1
3
5
7
9
11
...
47
49
```

Infinite while Loop

- We can create an infinite loop using a while loop by defining a condition that is always true.

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

int main() {

    // Infinite loop
    while (true) {
        cout << "gfg" << endl;
    }

    return 0;
}
```

Output

```
gfg
gfg
.
.
.
infinite times
```

C++ do...while Loop

- The do...while loop is a variant of the while loop with one important difference: the body of do...while loop is executed once before the condition is checked.

Its syntax is:

```
do {
   // body of loop;
}
while (condition);
```

Here,

- The body of the loop is executed at first. Then the `condition` is evaluated.
- If the `condition` evaluates to `true`, the body of the loop inside the `do` statement is executed again.
- The `condition` is evaluated once again.
- If the `condition` evaluates to `true`, the body of the loop inside the `do` statement is executed again.
- This process continues until the `condition` evaluates to `false`. Then the loop stops.

Syntax of the *do-while* Statement

A Loop Body with Several Statements:

```
do
{
        Statement_1
        Statement_2
          . . .
        Statement_Last
} while (Boolean_Expression);
```

body

Do not forget the final semicolon.

A Loop Body with a Single Statement:

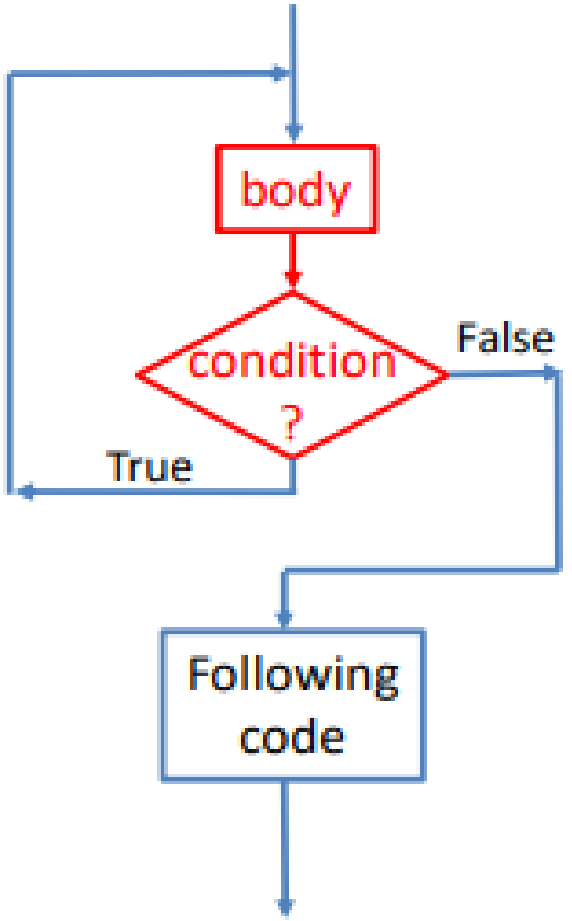
```
do
        Statement
while (Boolean_Expression);
```

body

Do-While Loop Syntax

```
do {
        // code in body of loop
} while( condition );
```

- Note required semi-colon
- If braces are omitted, then a single statement comprises the body of the loop. (Very rarely omitted.)



Example : Sum of Positive Numbers Only

- Here, the do...while loop continues until the user enters a negative number.
- When the number is negative, the loop terminates; the negative number is not added to the sum variable.

```
Enter a number: 6
Enter a number: 12
Enter a number: 7
Enter a number: 0
Enter a number: -2

The sum is 25
```

```
// program to find the sum of positive numbers
// If the user enters a negative number, the loop ends
// the negative number entered is not added to the sum

#include <iostream>
using namespace std;

int main() {
    int number = 0;
    int sum = 0;

    do {
        sum += number;

        // take input from the user
        cout << "Enter a number: ";
        cin >> number;
    }
    while (number >= 0);

    // display the sum
    cout << "\nThe sum is " << sum << endl;

    return 0;
}
```

Write a program to print sum of odd numbers ?

```cpp
#include <iostream>

using namespace std;

int main()
{
   int sum=0;;
   int i = 1;
   do {
   sum= sum+i;
   i+=2;
   }
   while (i < 40);
   cout << sum ;
     return 0;
}
```

```
400
Process returned 0 (0x0)   exec
Press any key to continue.
```

Infinite Loops

- Loops that never stop are infinite loops
- The loop body should contain a line that will eventually cause the boolean expression to become false
- Example: Print the odd numbers less than 12

```
x = 1;
while (x != 12)
    {
       cout << x << endl;
       x = x + 2;
    }
```

- Better to use this comparison: while (x < 12)

C++ break

It is used to "jump out" of a switch statement.

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

```cpp
#include <iostream>

using namespace std;

int main()
{

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


    return 0;
}
```

```
D:\app\ssss\bin\Debug\ssss.exe
0
1
2
3

Process returned 0 (0x0)   exe
Press any key to continue.
```

C++ continue

- The continue statement breaks one iteration (in the loop), if a specified condition occurs, and continues with the next iteration in the loop.
- This example skips the value of 4:

```
#include <iostream>

using namespace std;

int main()
{

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

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

Process returned 0 (
Press any key to con
```

Break	**continue**
• Used to **exit (stop) the loop immediately**, even if the loop condition is still true. • The program continues executing the code **after** the loop	• Used to **skip** the rest of the loop body for the current iteration and move to the next one. • The loop does not stop; it just skips one turn.

C++ For Loop

When you know exactly how many times you want to loop through a block of code, use the `for` loop instead of a `while` loop:

Syntax

```
for (statement 1; statement 2; statement 3) {
  // code block to be executed
}
```

Statement 1 is executed (one time) before the execution of the code block.

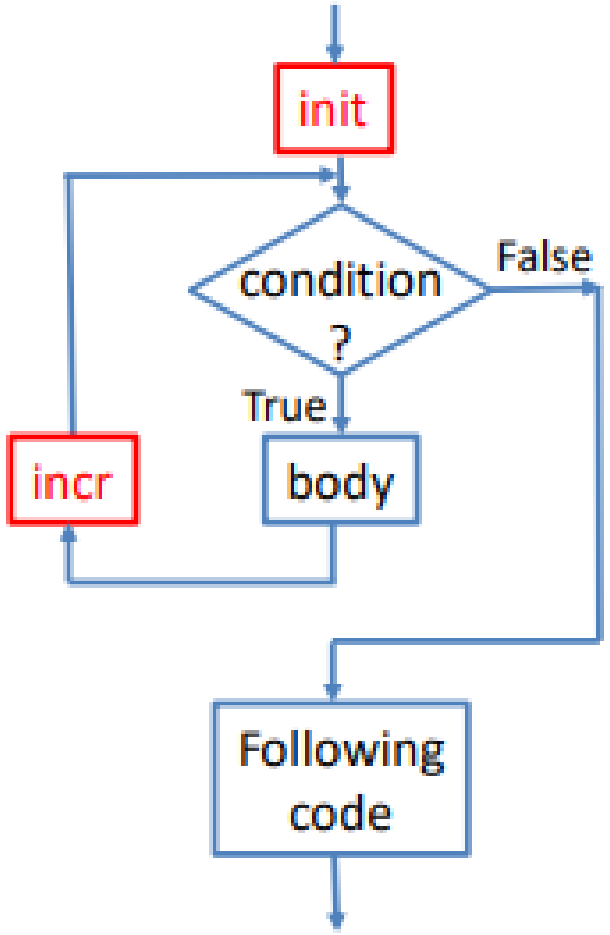
Statement 2 defines the condition for executing the code block.

Statement 3 is executed (every time) after the code block has been executed.

For Loop Syntax

```
for( init; condition; incr ) {
        // code in body of loop
}
```

- If braces are omitted, then a single statement comprises the body of the loop.
- Note "incr" always happens **after** executing the body.



Write a c++ program to print world 4 times?

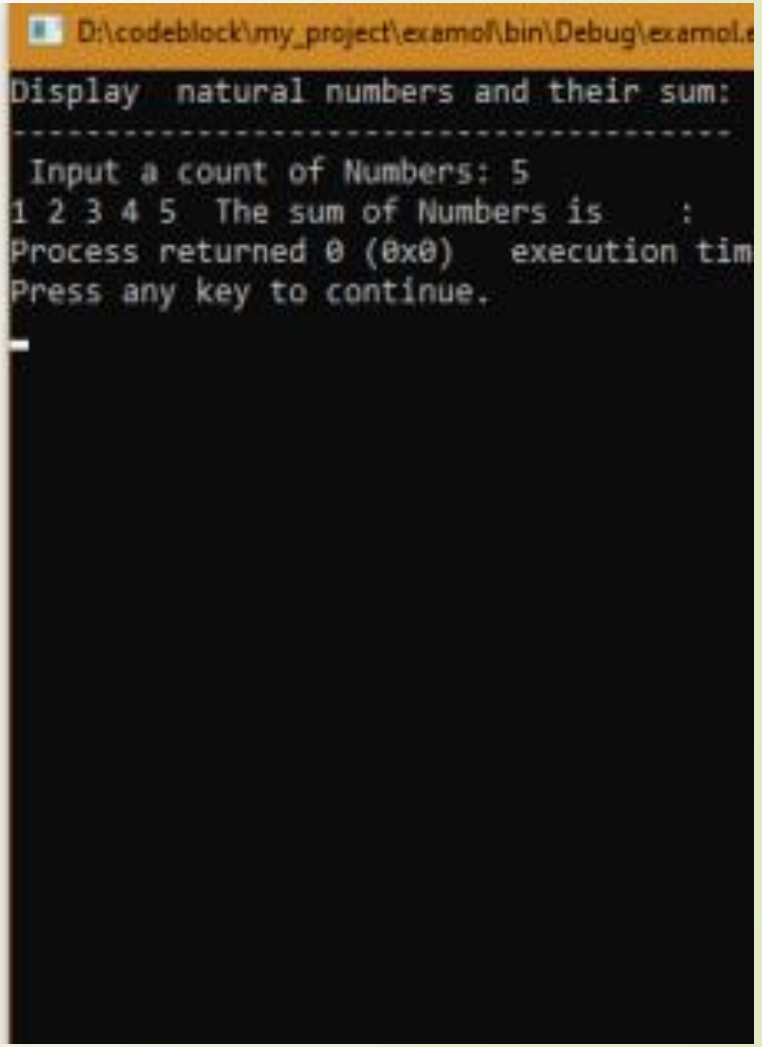
main.cpp

```cpp
#include <iostream>

using namespace std;

int main()
{

    for(int i=0;i<5;i++){

     cout<<"World\t";
    }


     return 0;
}

```

```
D:\codeblock\my_project\examol\bin\Debug\examol.e
World   World   World   World   World
Process returned 0 (0x0)   execution time
Press any key to continue.
```

Write a program in C++ to display n of numbers and their sum?

```cpp
#include <iostream>

using namespace std;

int main()
{

   int n,i,sum=0;
    cout << "Display  natural numbers and their sum:\n";
    cout << "-------------------------------------\n";
    cout << " Input a count of Numbers: ";
    cin>> n;

    for (i = 1; i <= n; i++)
    {
        cout << i << " ";
        sum=sum+i;
    }

    cout << " The sum of Numbers is     : ";
    return 0;
}
```



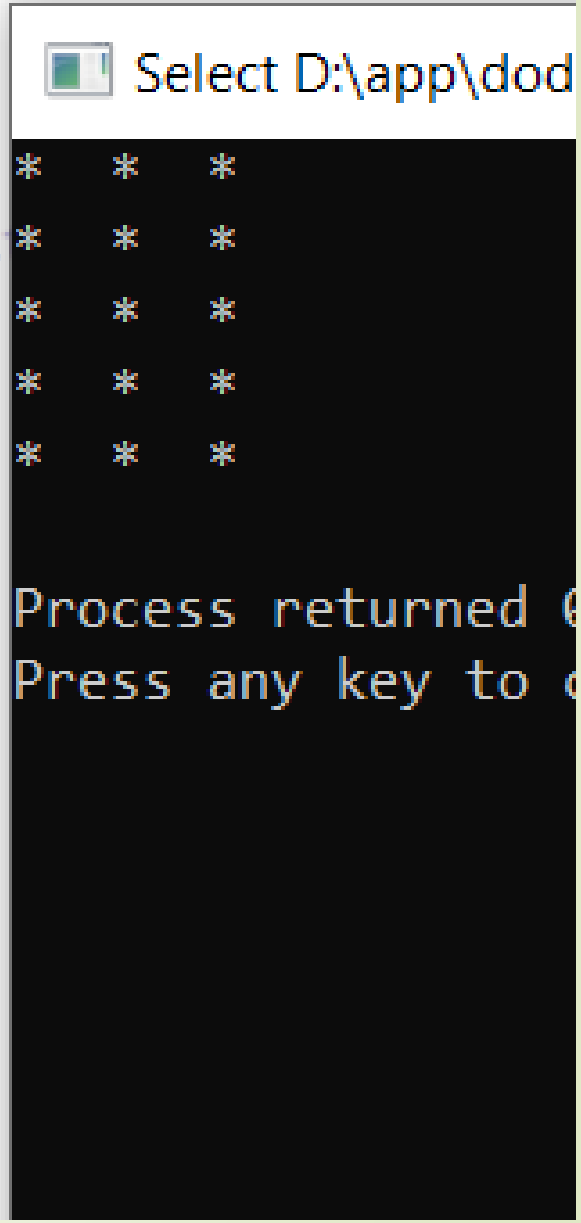
C++ Nested Loop

- A loop within another loop is called a nested loop.
- Suppose we want to loop through each day of a week for 3 weeks.

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

int main() // Main function where the execution of
{
  int weeks = 3, days_in_week = 7;

    for (int i = 1; i <= weeks; ++i) {
        cout << "Week: " << i << endl;

        for (int j = 1; j <= days_in_week; ++j) {
            cout << "    Day:" << j << endl;
        }
    }

   return 0;

}
```

```
D:\app\dody\bin\Debug\do
Week: 1
    Day:1
    Day:2
    Day:3
    Day:4
    Day:5
    Day:6
    Day:7
Week: 2
    Day:1
    Day:2
    Day:3
    Day:4
    Day:5
    Day:6
    Day:7
Week: 3
    Day:1
    Day:2
    Day:3
    Day:4
    Day:5
    Day:6
    Day:7
```

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

int main() // Main function where the execu
{
   int rows = 5;
    int columns = 3;

    for (int i = 1; i <= rows; ++i) {
        for (int j = 1; j <= columns; ++j) {
            cout << "*   ";
        }
        cout << endl;
    }
```



Write a program in C++ to check whether a number is prime or not?

```cpp
int main()
{

    int num1, ctr = 0;
    cout << "3 Check whether a number is prime or not:\n";
    cout << "----------------------------------------\n";
    cout << " Input a number to check prime or not: ";
    cin>> num1;
    for (int a = 1; a <= num1; a++)
    {
        if (num1 % a == 0)
        {
            ctr++;
        }
    }
    if (ctr == 2)
    {
        cout << " The entered number is a prime number. \n";
    }
    else {
        cout << " The number you entered is not a prime number.
    }
```

```
D:\codeblock\my_project\examol\bin\Debug\examol.exe
3 Check whether a number is prime or not:
----------------------------------------
 Input a number to check prime or not: 11
 The entered number is a prime number.

Process returned 0 (0x0)   execution time : 36
Press any key to continue.
```

Write a program in C++ to find the factorial of a number?

```cpp
#include <iostream>

using namespace std;

int main()
{

   int num1,factorial=1;
    cout << "Find the factorial of a number:\n";
    cout << "-----------------------------------\n";
    cout << " Input a number to find the factorial: ";
    cin>> num1;
    for(int a=1;a<=num1;a++)
    {
        factorial=factorial*a;
    }
    cout<<" The factorial of the given number is: "<<factorial<<endl;
    return 0;
}
```

```
D:\codeblock\my_project\examol\bin\Debug\examol.exe
Find the factorial of a number:
-----------------------------------
 Input a number to find the factorial: 10
 The factorial of the given number is: 3628800

Process returned 0 (0x0)   execution time : 9.520 s
Press any key to continue.
```

Write a program in C++ to display the cube of the number up to given an integer?

```cpp
#include <iostream>

using namespace std;

int main()
{

   int i, ctr, cub;

    cout << " Display the cube of the numbers up to a given integer:\n";
    cout << "-------------------------------------------------------\n";
    cout << "Input the number of terms : ";
    cin >> ctr;
    for (i = 1; i <= ctr; i++)
    {
        cub = i * i * i;
        cout << "Number is : " << i << " and the cube of " << i << " is: " << cub << endl;
    }
    return 0;
```

```
D:\codeblock\my_project\examol\bin\Debug\examol.exe
 Display the cube of the numbers up to a given integer:
-------------------------------------------------------
Input the number of terms :  6
Number is : 1 and the cube of 1 is: 1
Number is : 2 and the cube of 2 is: 8
Number is : 3 and the cube of 3 is: 27
Number is : 4 and the cube of 4 is: 64
Number is : 5 and the cube of 5 is: 125
Number is : 6 and the cube of 6 is: 216
```

References

- John T. Bell, C/C++ Programming for Engineers: Loops, Department of Computer Science University of Illinois, Chicago 2018
- https://www.programiz.com/cpp-programming/nested-loops
- https://www.geeksforgeeks.org/cpp/cpp-while-loop/
- https://www.programiz.com/cpp-programming/do-while-loop

Thank you