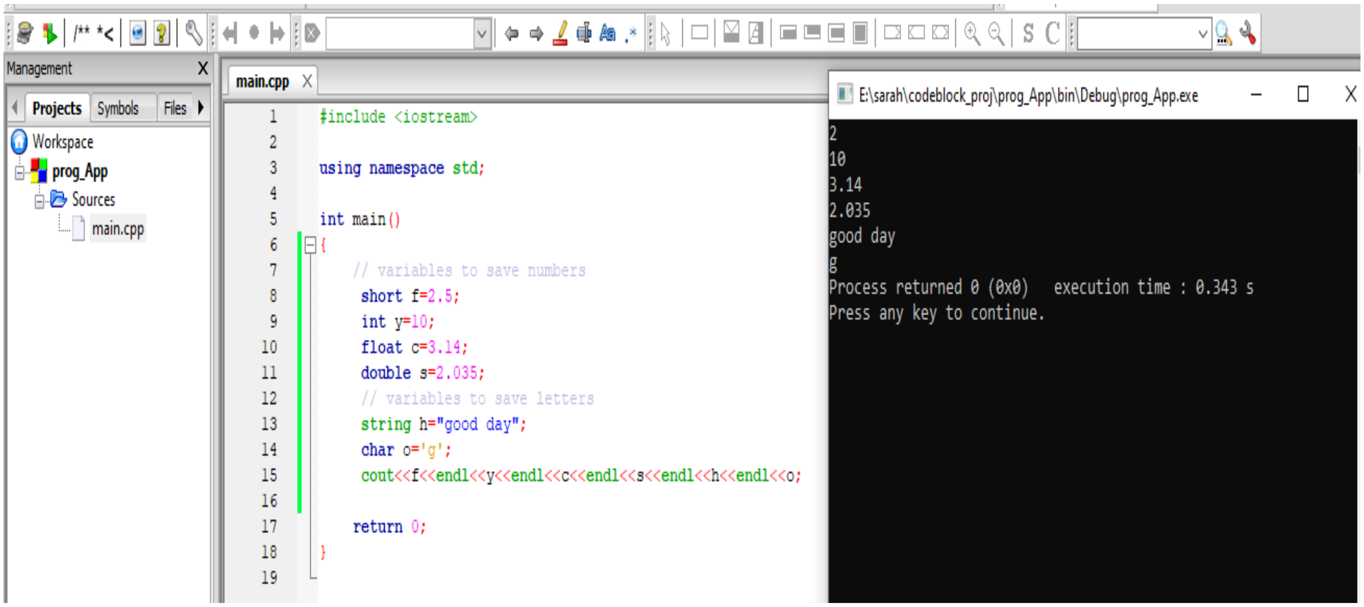
Exercises – chapter2

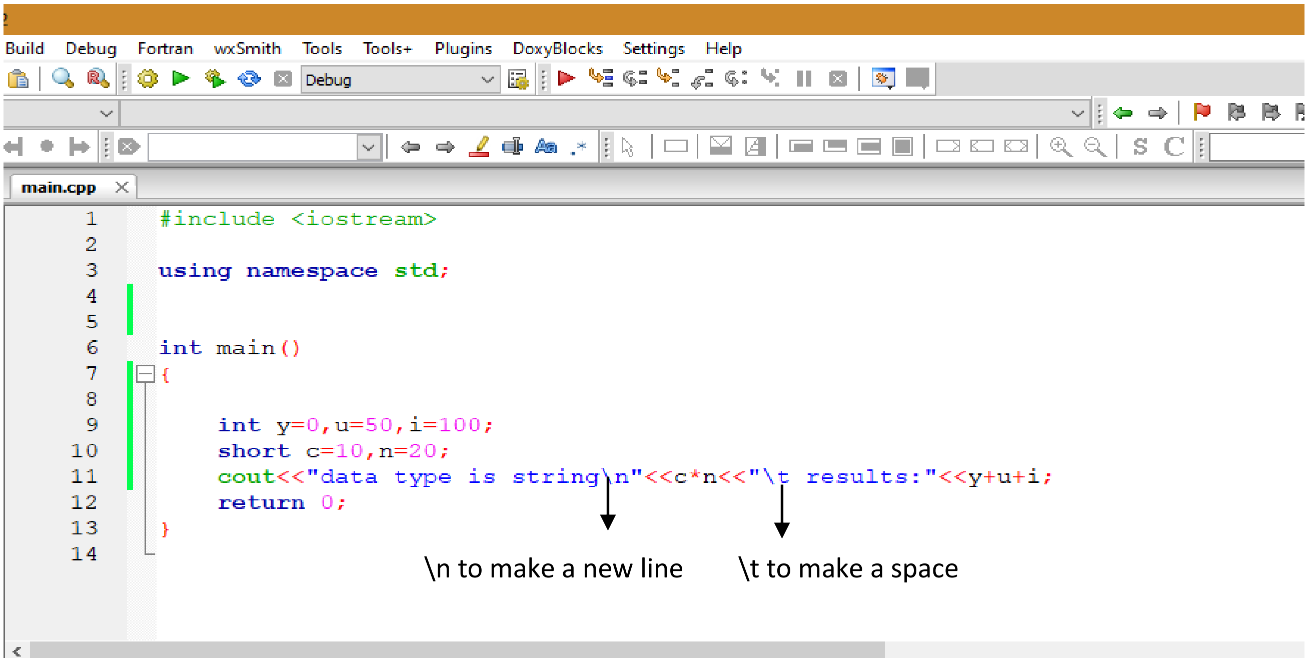
Variables and Assignments



```cpp
#include <iostream>

using namespace std;

int main()
{
    // variables to save numbers
     short f=2.5;
     int y=10;
     float c=3.14;
     double s=2.035;
     // variables to save letters
     string h="good day";
     char o='g';
     cout<<f<<endl<<y<<endl<<c<<endl<<s<<endl<<h<<endl<<o;

    return 0;
}
```

```
2
10
3.14
2.035
good day
g
Process returned 0 (0x0)   execution time : 0.343 s
Press any key to continue.
```

Variable declaration/initialization



```cpp
#include <iostream>

using namespace std;


int main()
{

    int y=0,u=50,i=100;
    short c=10,n=20;
    cout<<"data type is string\n"<<c*n<<"\t results:"<<y+u+i;
    return 0;
}
```

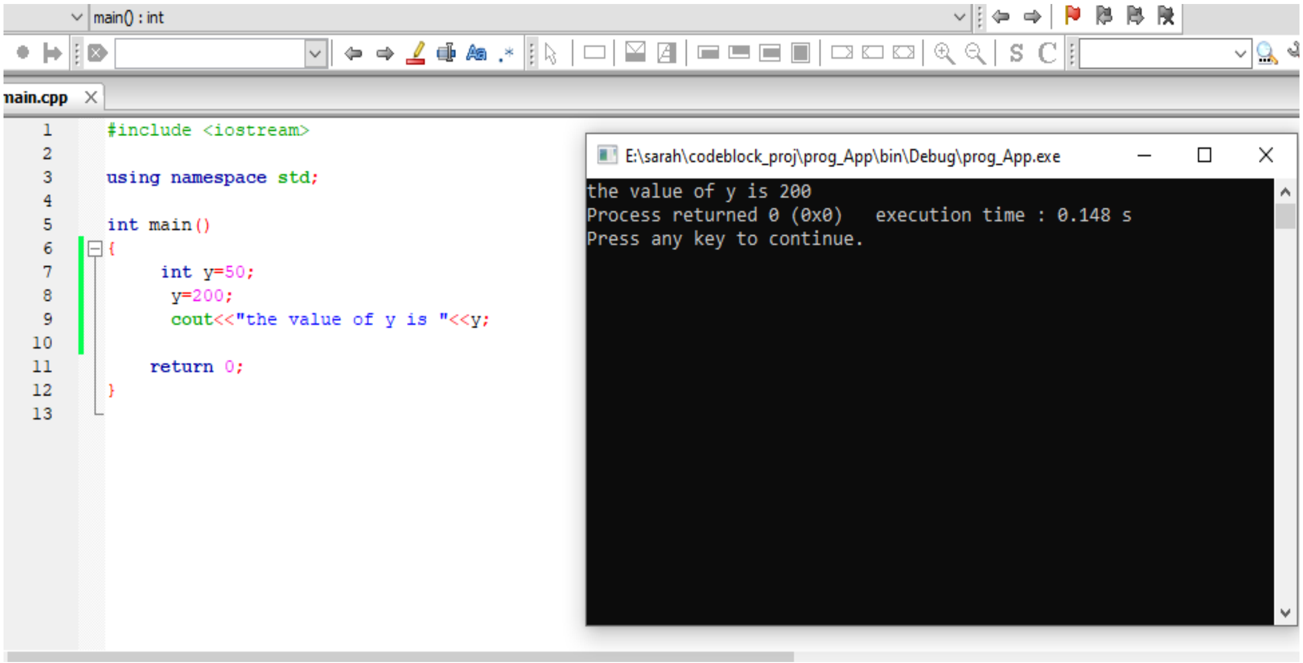
\n to make a new line   \t to make a space

Basic Data Types

The data type specifies the size and type of information the variable will store:

Data Type	Size	Description
int	4 bytes	Stores whole numbers, without decimals
float	4 bytes	Stores fractional numbers, containing one or more decimals. Sufficient for storing 7 decimal digits
double	8 bytes	Stores fractional numbers, containing one or more decimals. Sufficient for storing 15 decimal digits
boolean	1 byte	Stores true or false values
char	1 byte	Stores a single character/letter/number, or ASCII values

You can reassign variable:



```cpp
#include <iostream>

using namespace std;

int main()
{
     int y=50;
      y=200;
      cout<<"the value of y is "<<y;

    return 0;
}
```

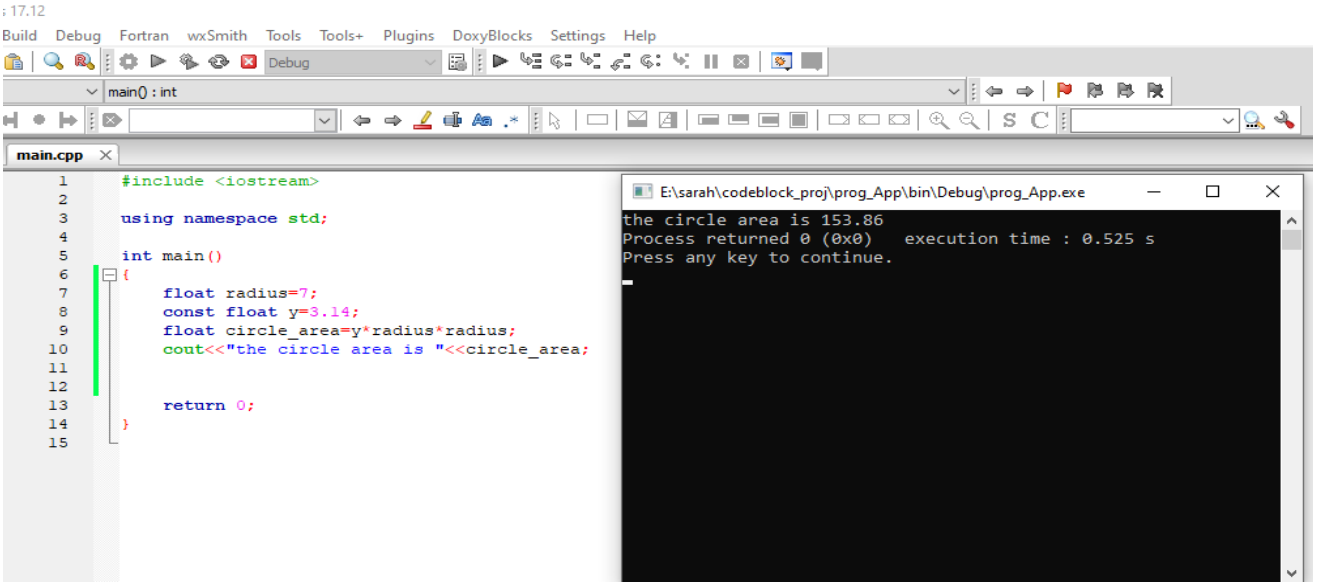
```
the value of y is 200
Process returned 0 (0x0)   execution time : 0.148 s
Press any key to continue.
```

Constants

- Used to define a variable whose value cannot be changed.
- Constant declaration syntax: const type constant name;

- Must declare and initialize constant in same time.
- Constant can’t reinitialize.

Example: Write C++ program to Compute circle area?



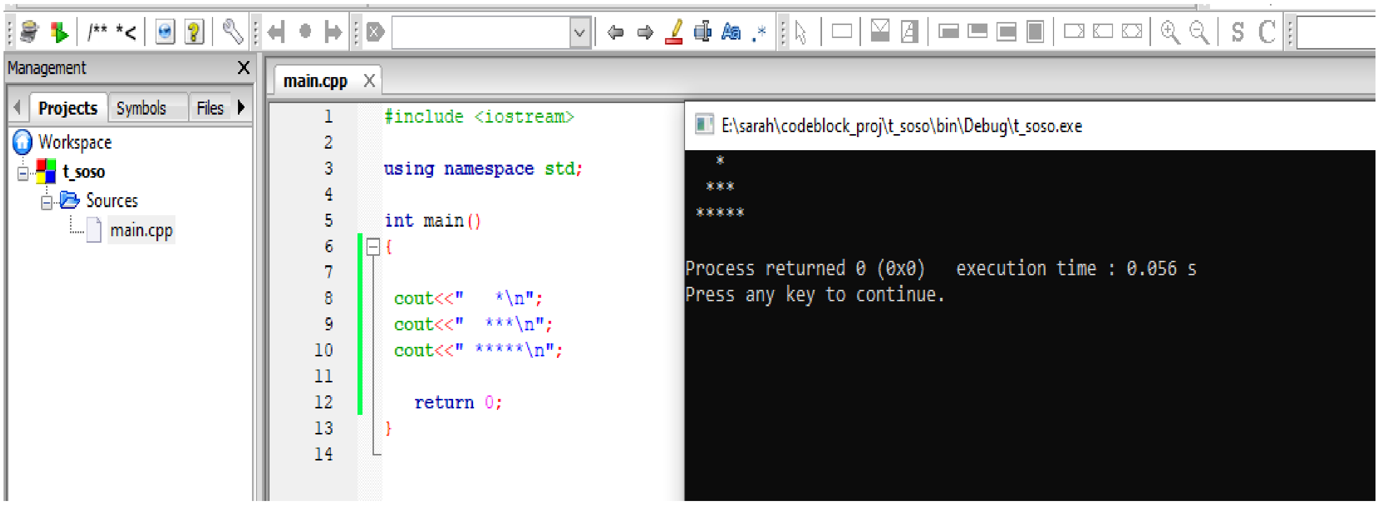
```
#include <iostream>

using namespace std;

int main()
{
    float radius=7;
    const float y=3.14;
    float circle_area=y*radius*radius;
    cout<<"the circle area is "<<circle_area;


    return 0;
}
```

E:\sarah\codeblock_proj\prog_App\bin\Debug\prog_App.exe

```
the circle area is 153.86
Process returned 0 (0x0)   execution time : 0.525 s
Press any key to continue.
```

Input and Output

Example for cout command

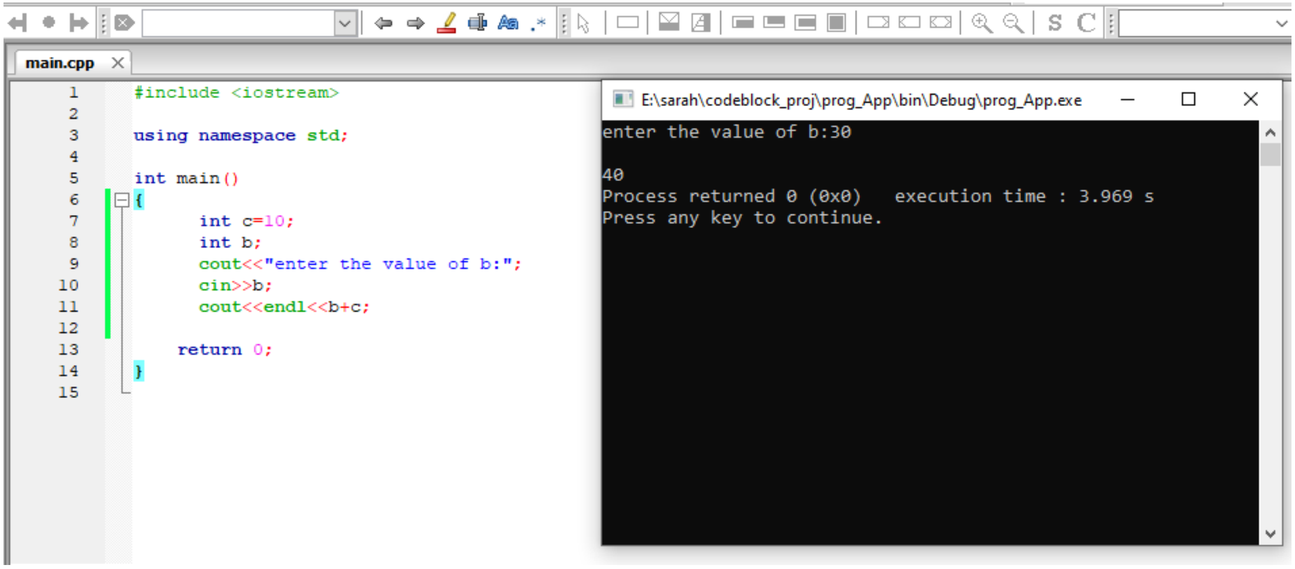


```
#include <iostream>

using namespace std;

int main()
{

 cout<<"   *\n";
 cout<<"  ***\n";
 cout<<" *****\n";

   return 0;
}
```

E:\sarah\codeblock_proj\t_soso\bin\Debug\t_soso.exe

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


Process returned 0 (0x0)   execution time : 0.056 s
Press any key to continue.
```

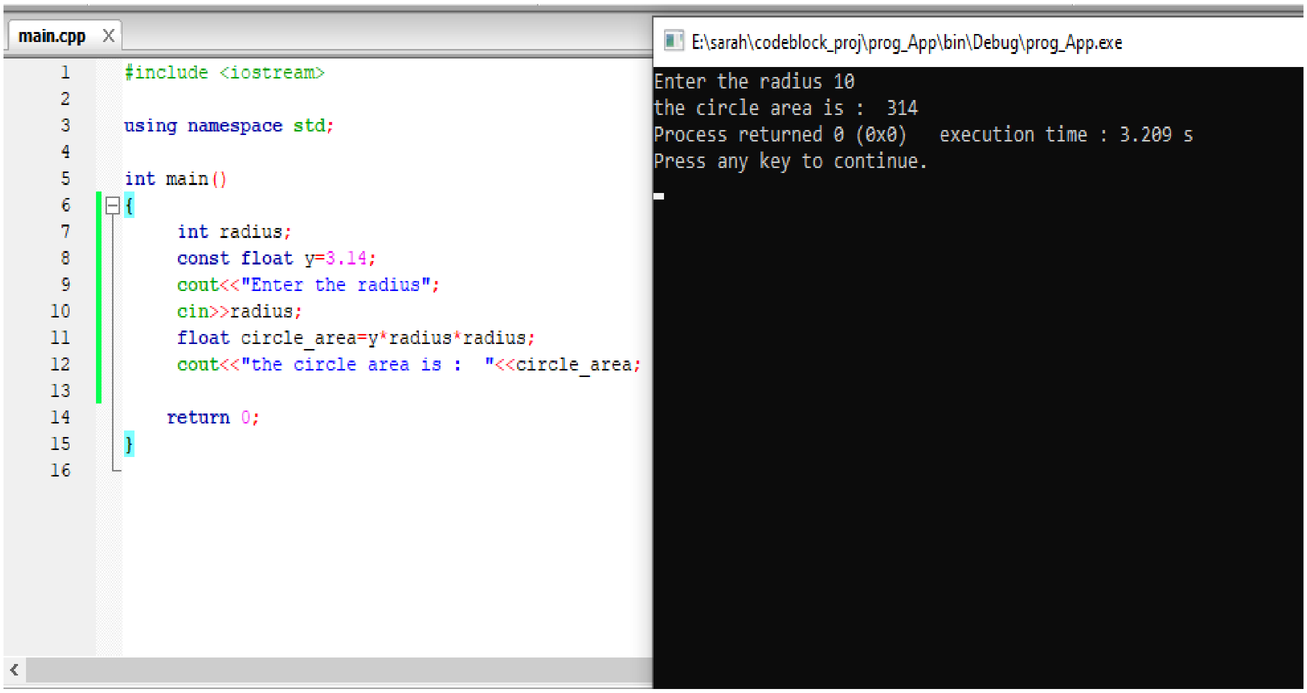
➨cin is a predefined variable that reads data from the keyboard with the extraction operator (>>).



```cpp
#include <iostream>

using namespace std;

int main()
{
      int c=10;
      int b;
      cout<<"enter the value of b:";
      cin>>b;
      cout<<endl<<b+c;

    return 0;
}
```

```
enter the value of b:30

40
Process returned 0 (0x0)   execution time : 3.969 s
Press any key to continue.
```

Example 2:Write c++ program to Compute circle area by cin command?

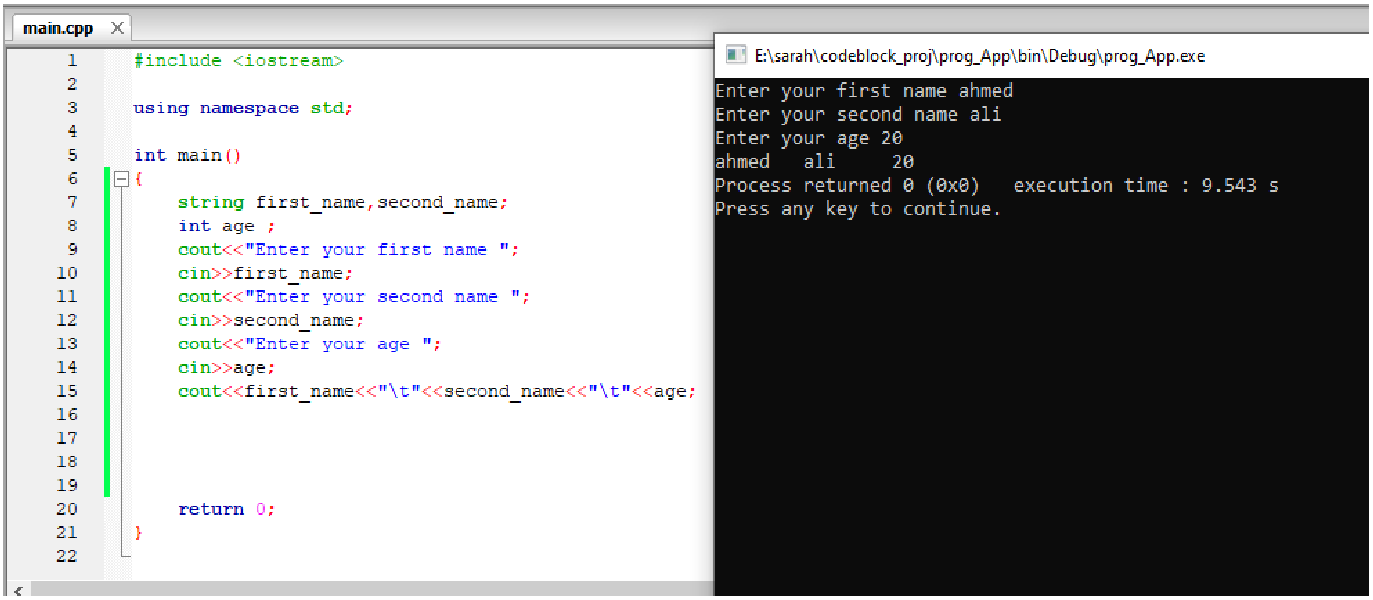


```cpp
#include <iostream>

using namespace std;

int main()
{
      int radius;
      const float y=3.14;
      cout<<"Enter the radius";
      cin>>radius;
      float circle_area=y*radius*radius;
      cout<<"the circle area is :  "<<circle_area;

    return 0;
}
```

```
Enter the radius 10
the circle area is :  314
Process returned 0 (0x0)   execution time : 3.209 s
Press any key to continue.
```

Write c++ program that ask the user to enter his/her name and age then print them in same line?

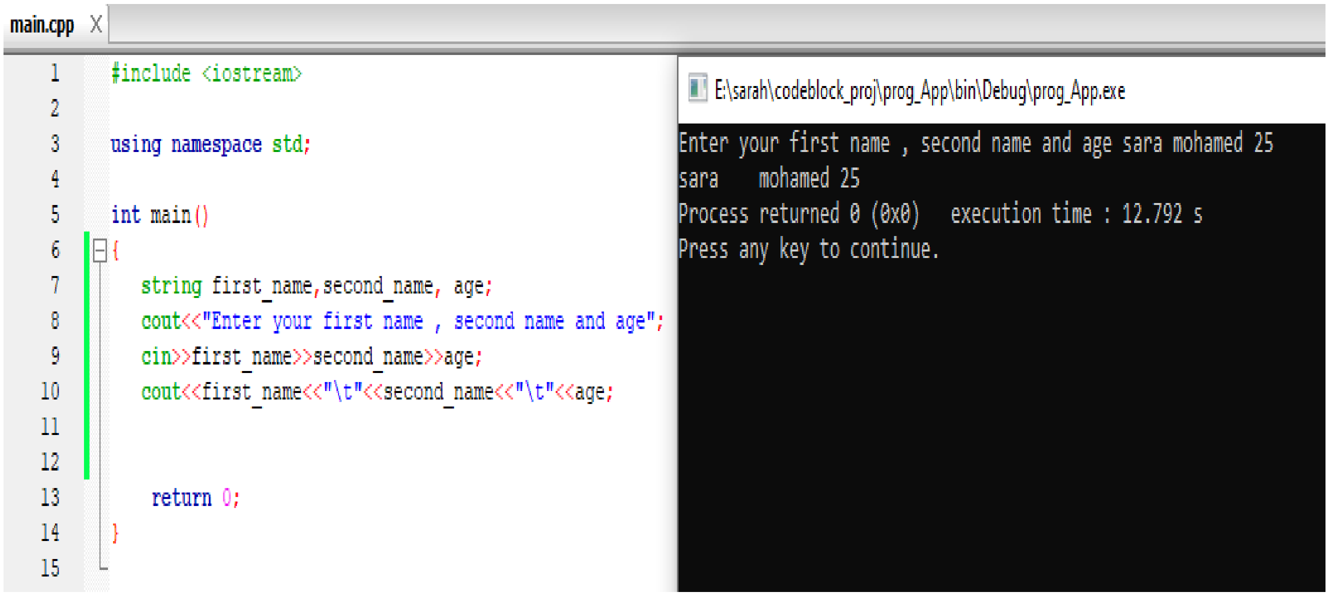
1. First method



```cpp
#include <iostream>

using namespace std;

int main()
{
    string first_name,second_name;
    int age ;
    cout<<"Enter your first name ";
    cin>>first_name;
    cout<<"Enter your second name ";
    cin>>second_name;
    cout<<"Enter your age ";
    cin>>age;
    cout<<first_name<<"\t"<<second_name<<"\t"<<age;




    return 0;
}

```

```
Enter your first name ahmed
Enter your second name ali
Enter your age 20
ahmed   ali     20
Process returned 0 (0x0)   execution time : 9.543 s
Press any key to continue.
```

2. Second method



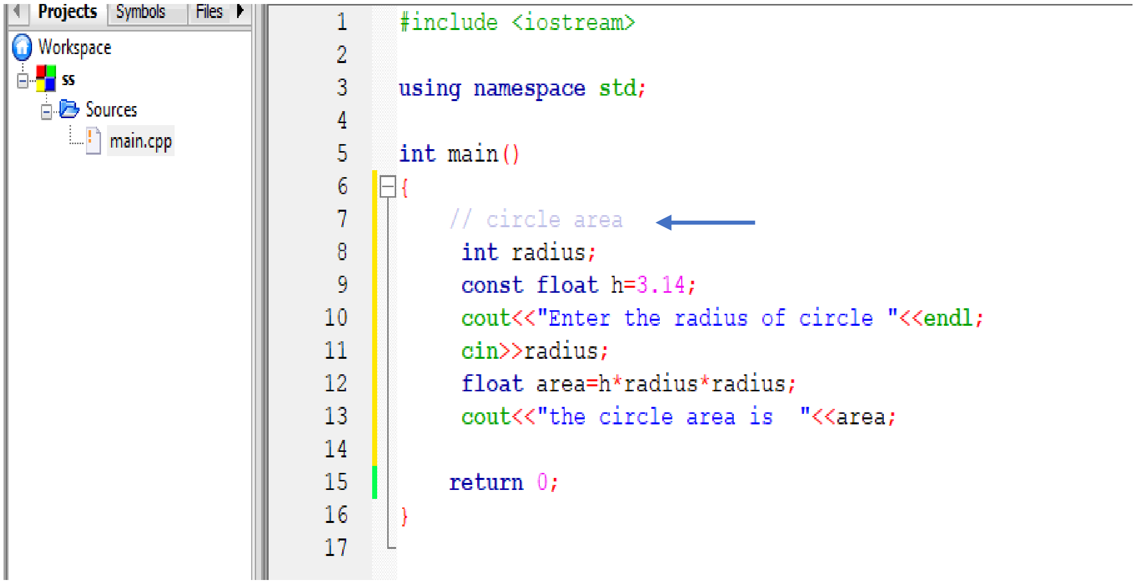
```cpp
#include <iostream>

using namespace std;

int main()
{
    string first_name,second_name, age;
    cout<<"Enter your first name , second name and age";
    cin>>first_name>>second_name>>age;
    cout<<first_name<<"\t"<<second_name<<"\t"<<age;


     return 0;
}

```

```
Enter your first name , second name and age sara mohamed 25
sara    mohamed 25
Process returned 0 (0x0)   execution time : 12.792 s
Press any key to continue.
```

Comments

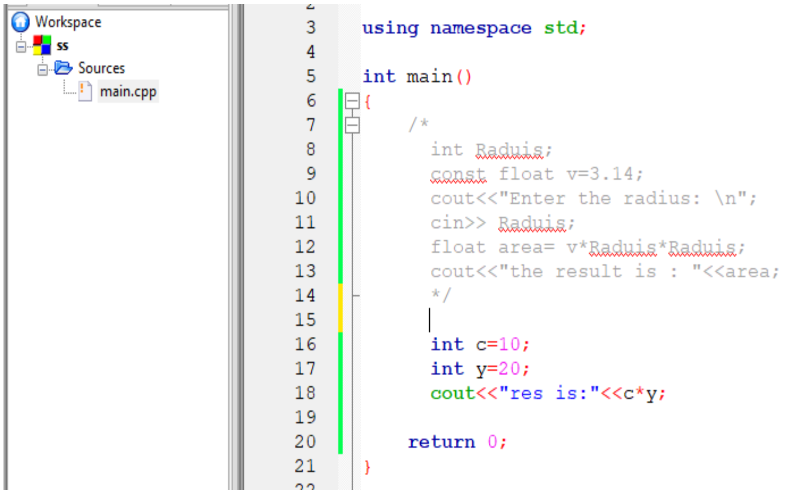
- Comments can be used to explain C++ code
- to make it more readable. It can also be used to prevent execution when testing alternative code.
- Comments can be **singled-lined or multi-lined**

Example for singled-lined comment



```
#include <iostream>

using namespace std;

int main()
{
    // circle area
     int radius;
     const float h=3.14;
     cout<<"Enter the radius of circle "<<endl;
     cin>>radius;
     float area=h*radius*radius;
     cout<<"the circle area is  "<<area;

    return 0;
}

```

Example for multi-lined comment



```
using namespace std;

int main()
{
    /*
     int Raduis;
     const float v=3.14;
     cout<<"Enter the radius: \n";
     cin>> Raduis;
     float area= v*Raduis*Raduis;
     cout<<"the result is : "<<area;
     */

     int c=10;
     int y=20;
     cout<<"res is:"<<c*y;

    return 0;
}
```

Operators

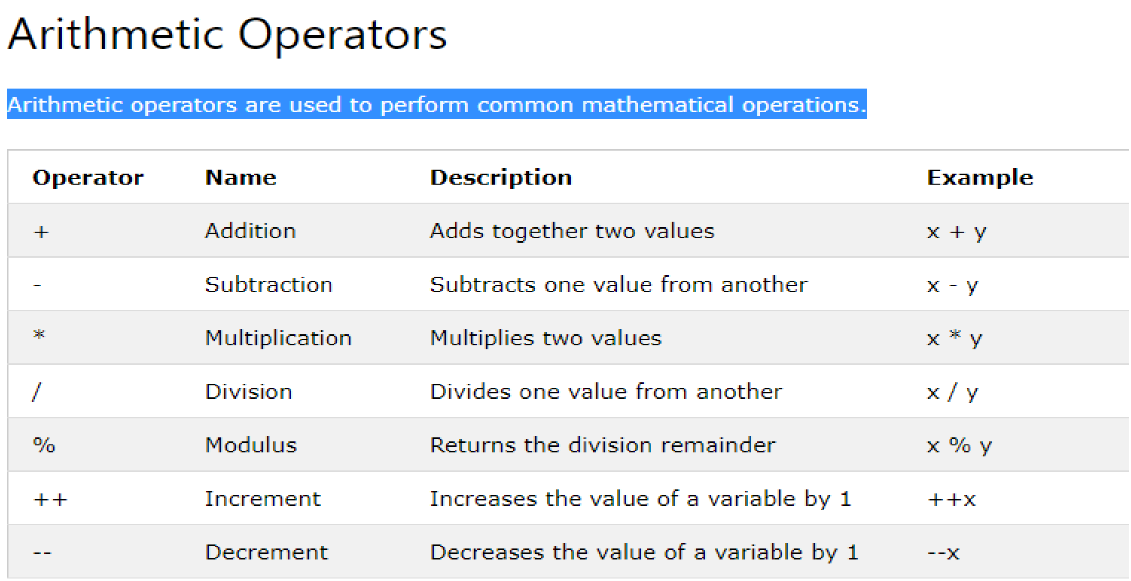
- Operators are used to perform operations on variables and values.

C++ divides the operators into the following groups:

- Arithmetic operators
- Assignment operators
- Comparison operators
- Logical operators

Arithmetic Operators

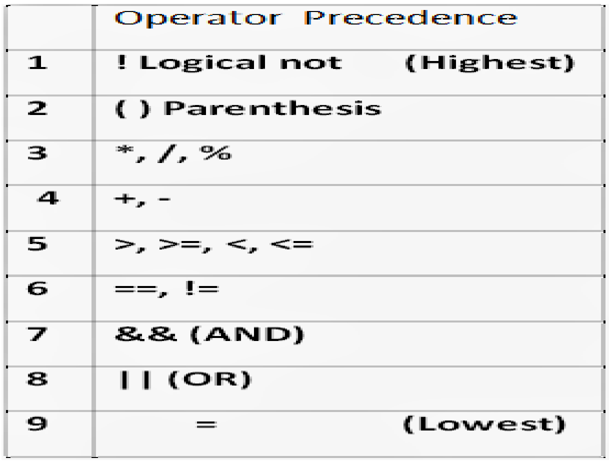
Arithmetic operators are used to perform common mathematical operations.



Arithmetic Operators

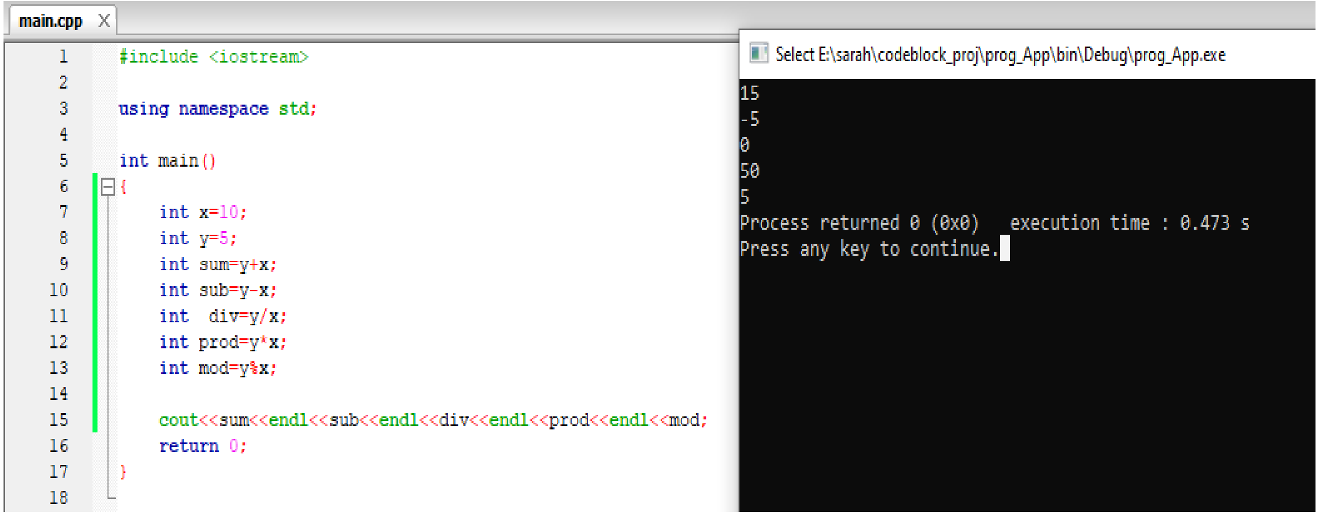
Arithmetic operators are used to perform common mathematical operations.

Operator	Name	Description	Example
+	Addition	Adds together two values	x + y
-	Subtraction	Subtracts one value from another	x - y
*	Multiplication	Multiplies two values	x * y
/	Division	Divides one value from another	x / y
%	Modulus	Returns the division remainder	x % y
++	Increment	Increases the value of a variable by 1	++x
--	Decrement	Decreases the value of a variable by 1	--x



	Operator Precedence
1	! Logical not (Highest)
2	( ) Parenthesis
3	*, /, %
4	+, -
5	>, >=, <, <=
6	==, !=
7	&& (AND)
8	|| (OR)
9	= (Lowest)

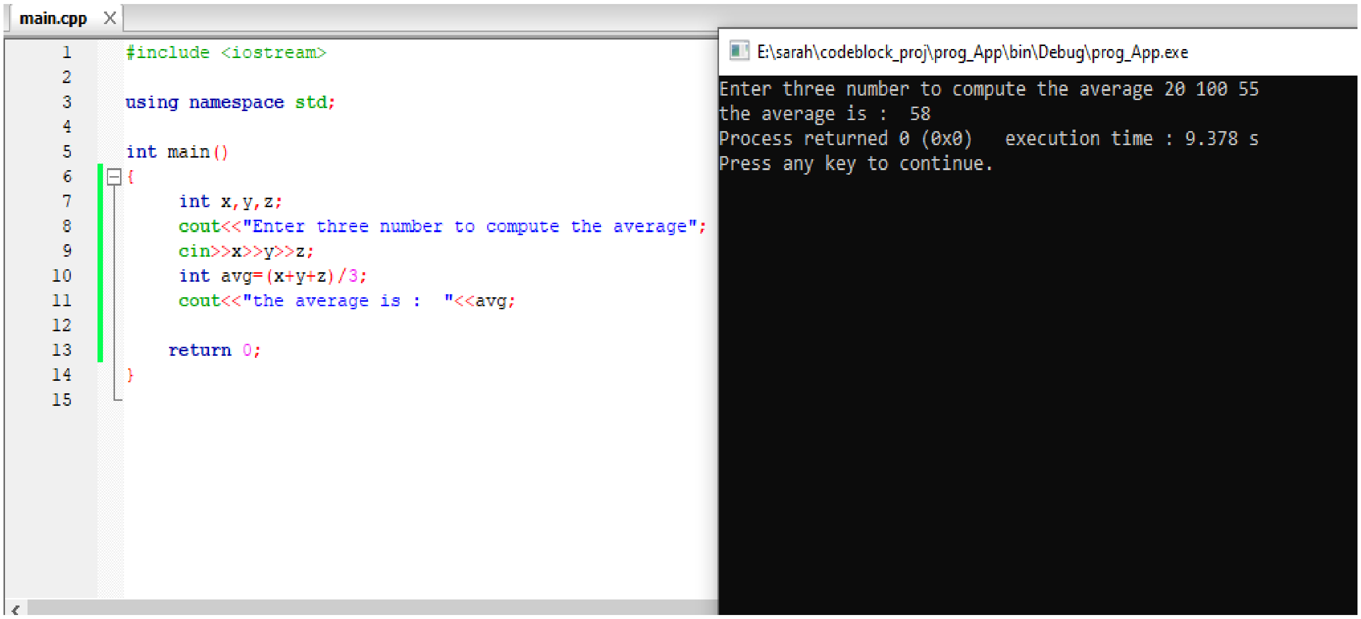
Example: Write a program in C++ to display various type or arithmetic operation ?


main.cpp
Select E:\sarah\codeblock_proj\prog_App\bin\Debug\prog_App.exe

```cpp
#include <iostream>

using namespace std;

int main()
{
    int x=10;
    int y=5;
    int sum=y+x;
    int sub=y-x;
    int  div=y/x;
    int prod=y*x;
    int mod=y%x;

    cout<<sum<<endl<<sub<<endl<<div<<endl<<prod<<endl<<mod;
    return 0;
}
```

```
15
-5
0
50
5
Process returned 0 (0x0)   execution time : 0.473 s
Press any key to continue.
```

Write c++ program that ask the user to enter 3 numbers and then program compute the average of these number and print the reslut on screen?

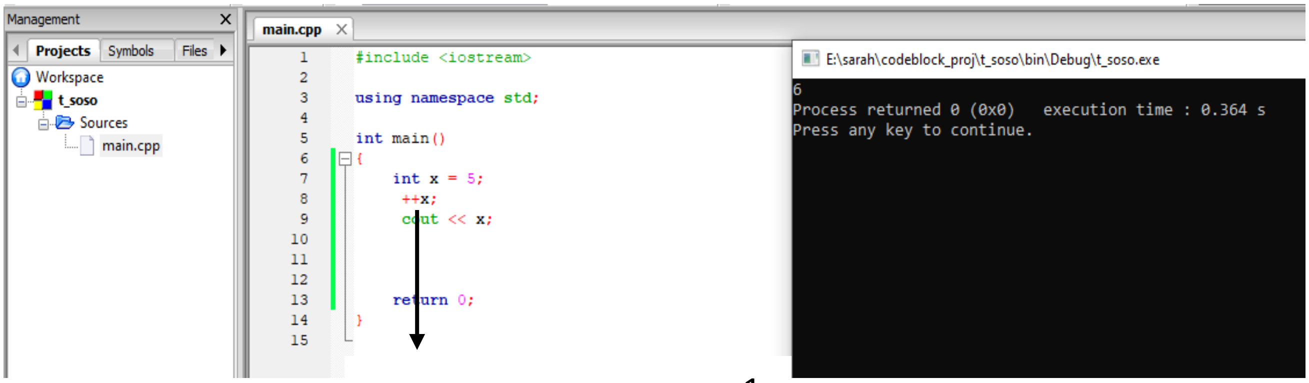


```cpp
#include <iostream>

using namespace std;

int main()
{
     int x,y,z;
     cout<<"Enter three number to compute the average";
     cin>>x>>y>>z;
     int avg=(x+y+z)/3;
     cout<<"the average is :  "<<avg;

    return 0;
}
```

```
Enter three number to compute the average 20 100 55
the average is :  58
Process returned 0 (0x0)   execution time : 9.378 s
Press any key to continue.
```

Increment and decrement

Increment: increase the variable value by 1.



```
#include <iostream>

using namespace std;

int main()
{
    int x = 5;
     ++x;
     cout << x;



    return 0;
}
```

```
6
Process returned 0 (0x0)   execution time : 0.364 s
Press any key to continue.
```

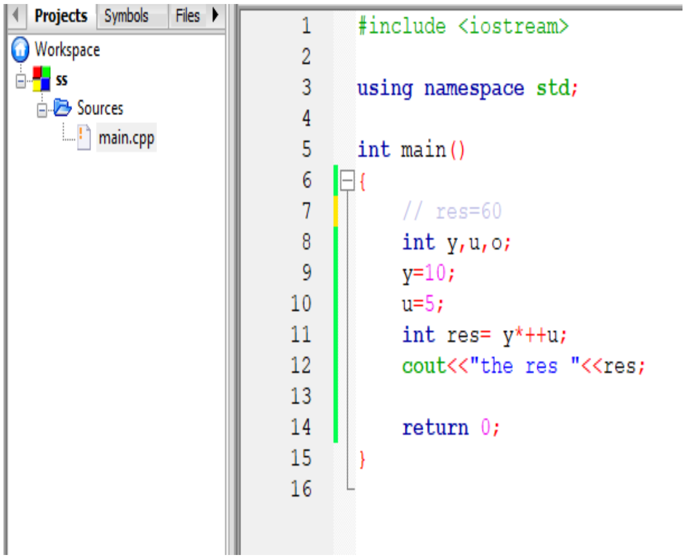
++x or x++ means x=x+1

Difference between ++x and x++:

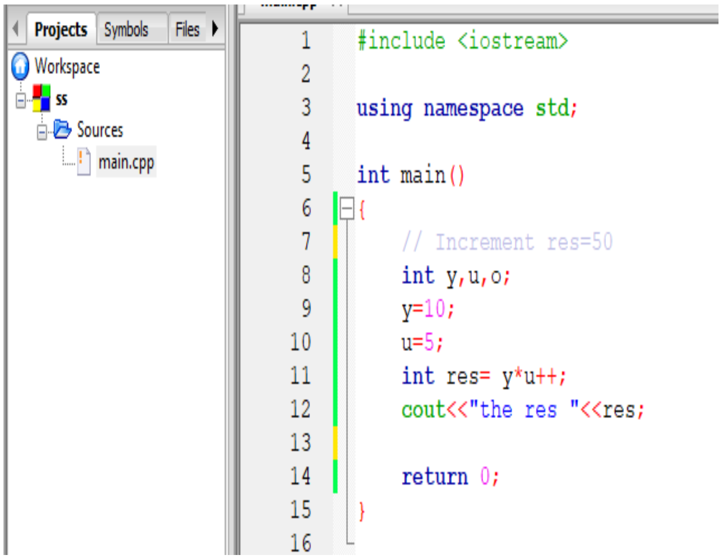
both ++x and x++ are used to increment variable x by 1.

The prime difference is that:

- ++x pre-increment operator uses the principle ‘change-then-use’.
- x++ post-increment operator uses the principle ‘use-then-change’**.**



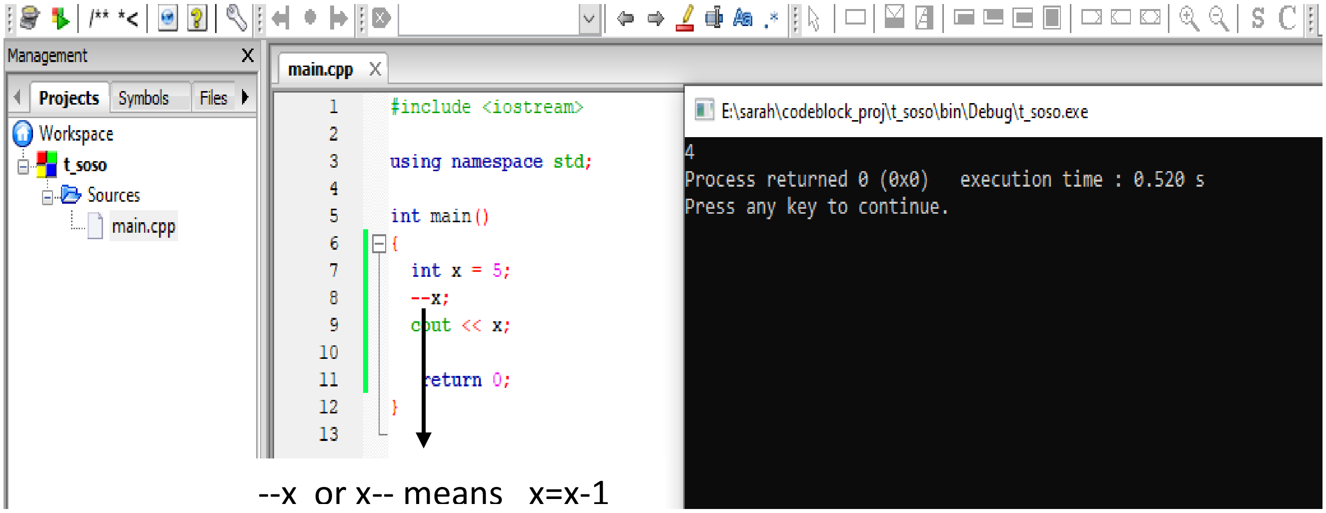
```
#include <iostream>

using namespace std;

int main()
{
    // res=60
    int y,u,o;
    y=10;
    u=5;
    int res= y*++u;
    cout<<"the res "<<res;

    return 0;
}
```



```
#include <iostream>

using namespace std;

int main()
{
    // Increment res=50
    int y,u,o;
    y=10;
    u=5;
    int res= y*u++;
    cout<<"the res "<<res;

    return 0;
}
```

Decrement

- ➨ decrease the variable value by 1

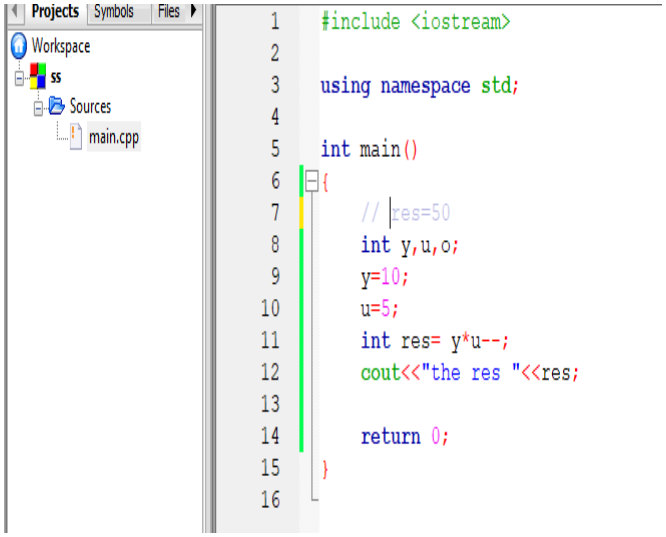


```cpp
#include <iostream>

using namespace std;

int main()
{
    int x = 5;
    --x;
    cout << x;

    return 0;
}
```

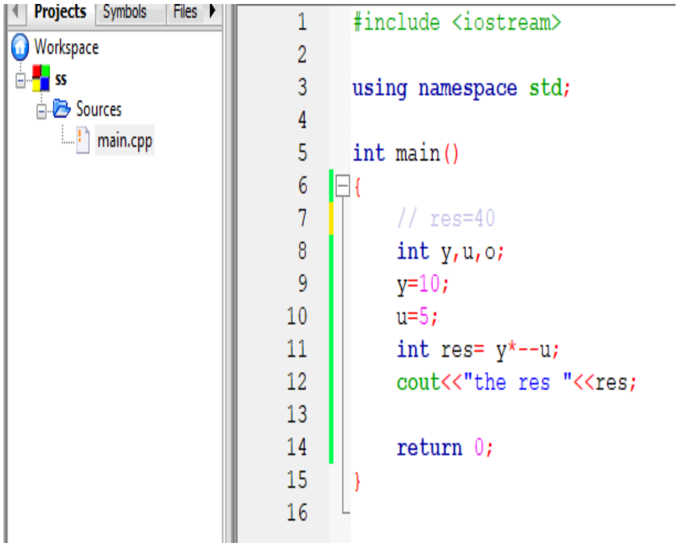
```
4
Process returned 0 (0x0)   execution time : 0.520 s
Press any key to continue.
```

--x or x-- means x=x-1

Difference between --x and x--



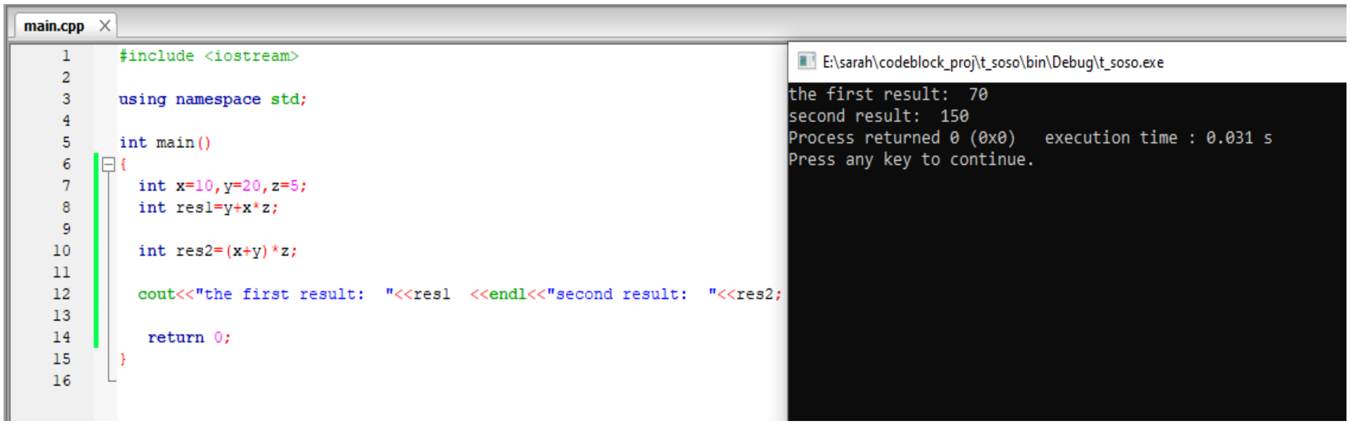
```cpp
#include <iostream>

using namespace std;

int main()
{
    // res=50
    int y,u,o;
    y=10;
    u=5;
    int res= y*u--;
    cout<<"the res "<<res;

    return 0;
}
```



```cpp
#include <iostream>

using namespace std;

int main()
{
    // res=40
    int y,u,o;
    y=10;
    u=5;
    int res= y*--u;
    cout<<"the res "<<res;

    return 0;
}
```

Precedence rules for operators are the same as used in your algebra classes.

Example:



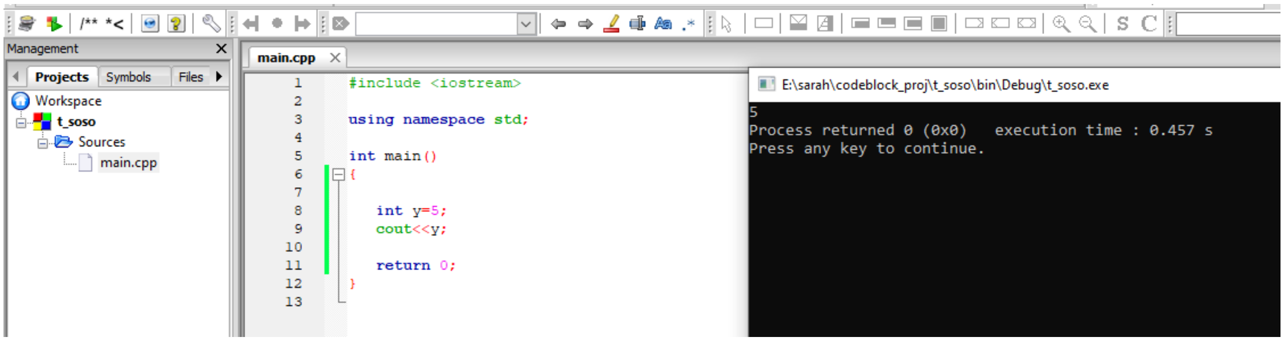
```cpp
#include <iostream>

using namespace std;

int main()
{
   int x=10,y=20,z=5;
   int res1=y+x*z;

   int res2=(x+y)*z;

   cout<<"the first result:  "<<res1  <<endl<<"second result:  "<<res2;

    return 0;
}
```

```
the first result:  70
second result:  150
Process returned 0 (0x0)   execution time : 0.031 s
Press any key to continue.
```

Operator Shorthand

Assignment operators are used to assign values to variables.

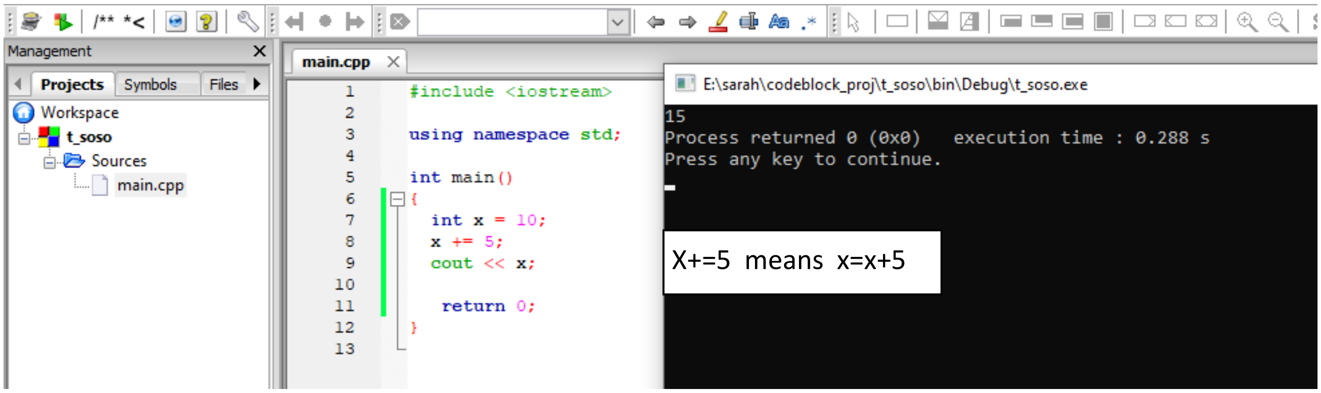
In the example below, we use the **assignment** operator (=) to assign the value **10** to a variable called **x**:



```cpp
#include <iostream>

using namespace std;

int main()
{

    int y=5;
    cout<<y;

    return 0;
}
```

```
5
Process returned 0 (0x0)   execution time : 0.457 s
Press any key to continue.
```

The **addition assignment** operator (+=) adds a value to a variable:

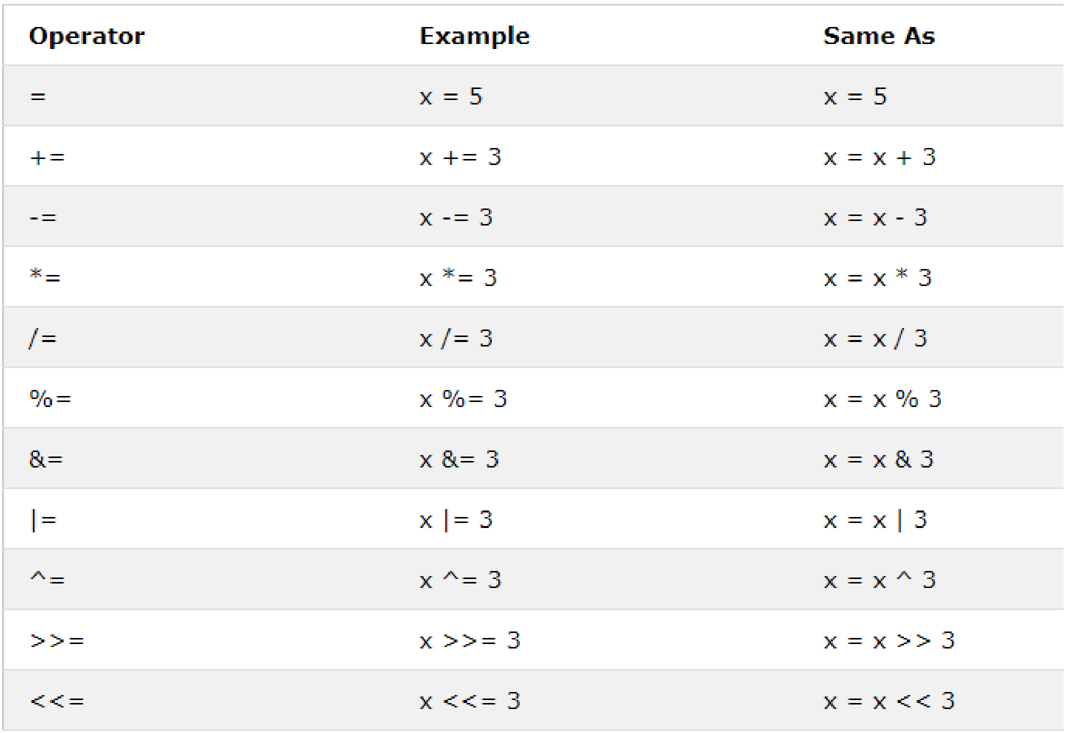


```cpp
#include <iostream>

using namespace std;

int main()
{
   int x = 10;
   x += 5;
   cout << x;

     return 0;
}
```

```
15
Process returned 0 (0x0)   execution time : 0.288 s
Press any key to continue.
```

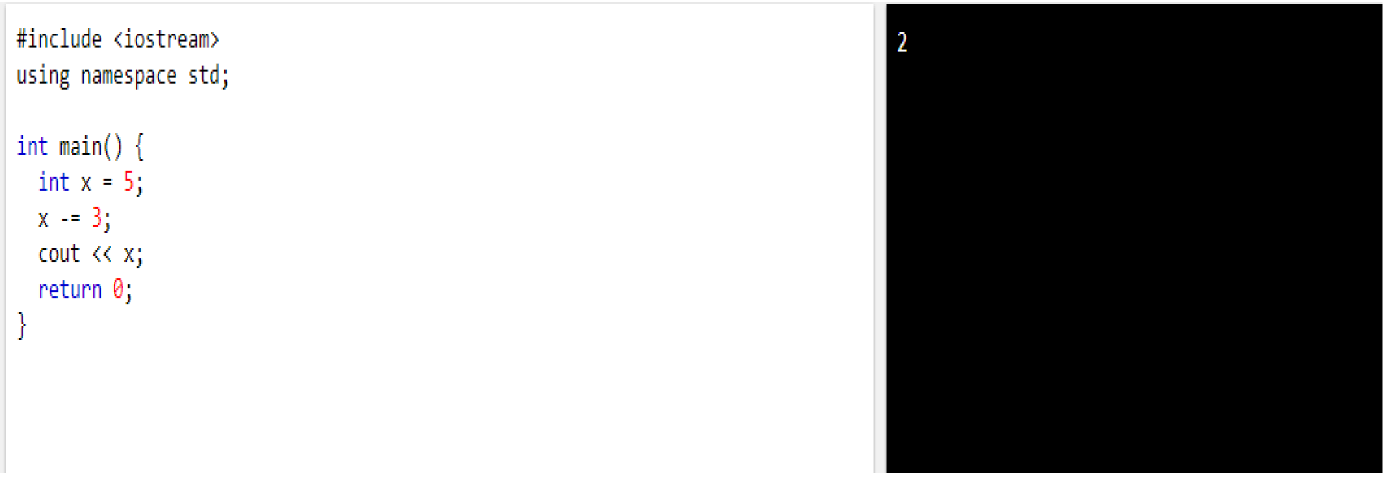
X+=5 means x=x+5

A list of all assignment operators:



Operator	Example	Same As
=	x = 5	x = 5
+=	x += 3	x = x + 3
-=	x -= 3	x = x - 3
*=	x *= 3	x = x * 3
/=	x /= 3	x = x / 3
%=	x %= 3	x = x % 3
&=	x &= 3	x = x & 3
|=	x |= 3	x = x | 3
^=	x ^= 3	x = x ^ 3
>>=	x >>= 3	x = x >> 3
<<=	x <<= 3	x = x << 3

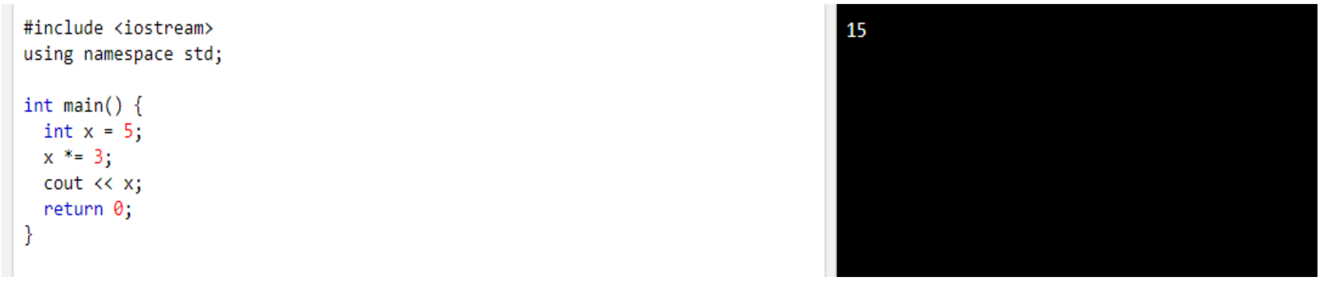
Example 1:



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

int main() {
  int x = 5;
  x -= 3;
  cout << x;
  return 0;
}
```

```
2
```

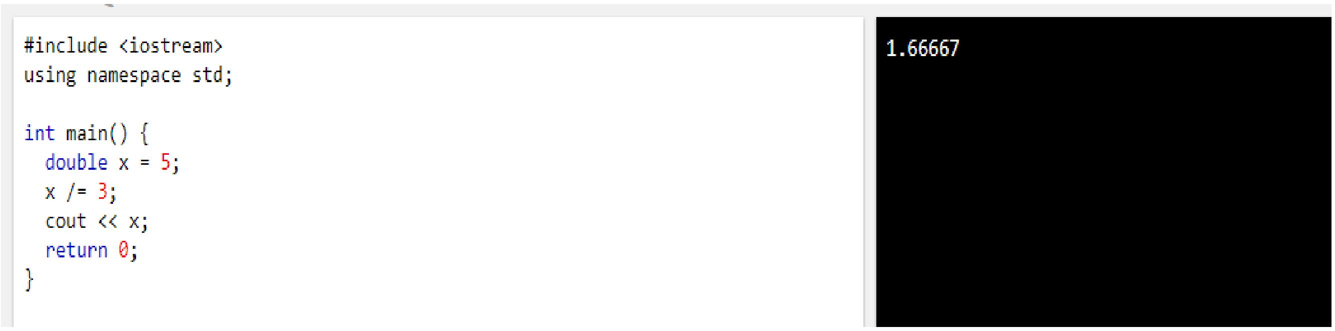
Example 2:



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

int main() {
  int x = 5;
  x *= 3;
  cout << x;
  return 0;
}
```

```
15
```

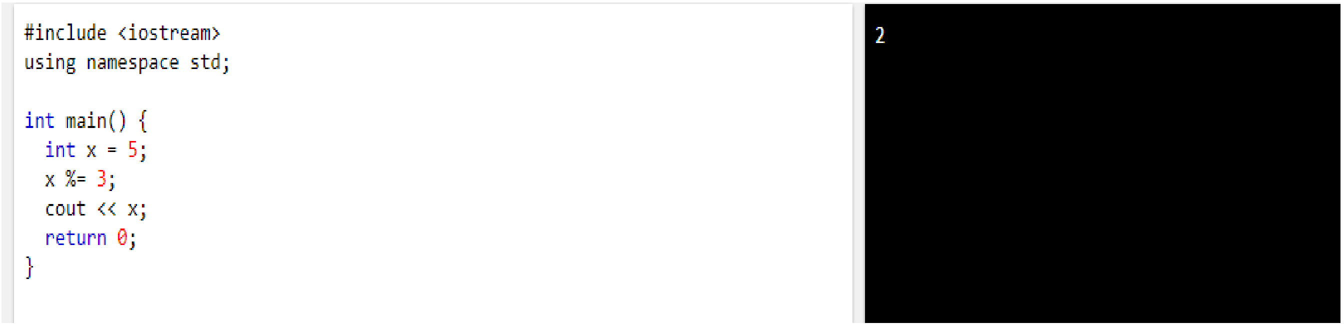
Example 3:



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

int main() {
  double x = 5;
  x /= 3;
  cout << x;
  return 0;
}
```

```
1.66667
```

Example 4:



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

int main() {
  int x = 5;
  x %= 3;
  cout << x;
  return 0;
}
```

```
2
```

Comparison Operators

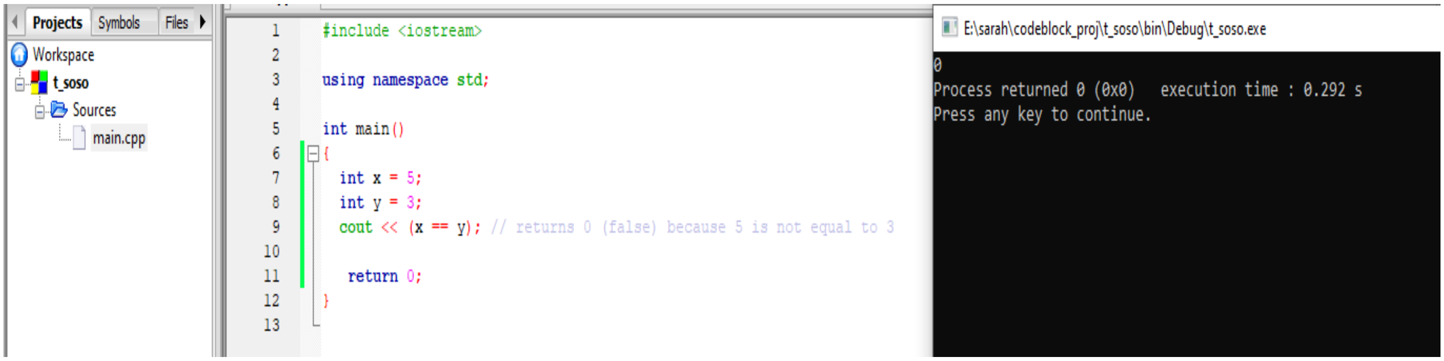
Comparison operators are used to compare two values.

Note: The return value of a comparison is either true (1) or false (0).

A list of all comparison operators:

Operator	Name	Example
==	Equal to	x == y
!=	Not equal	x != y
>	Greater than	x > y
<	Less than	x < y
>=	Greater than or equal to	x >= y
<=	Less than or equal to	x <= y

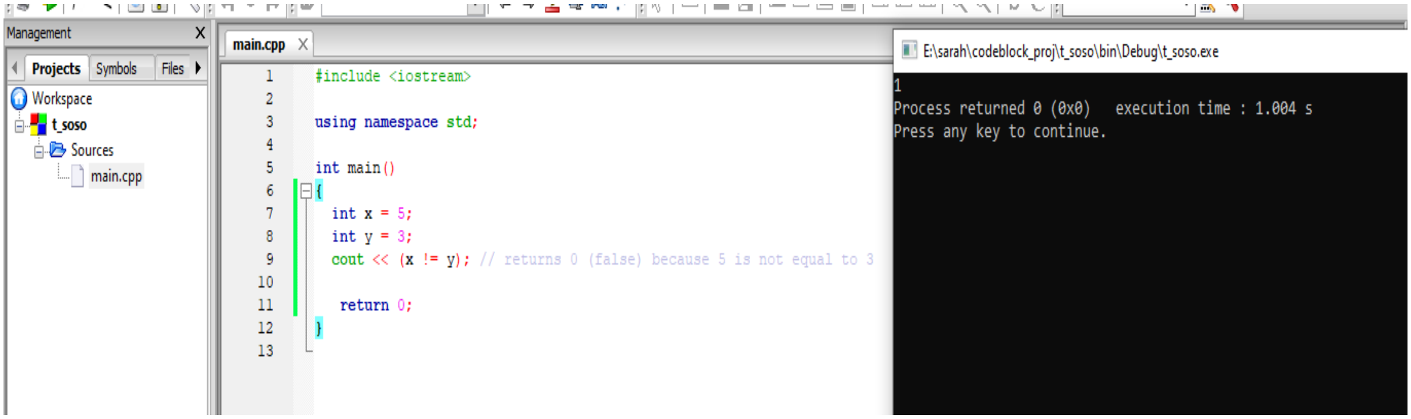
Example1:



```cpp
#include <iostream>

using namespace std;

int main()
{
  int x = 5;
  int y = 3;
  cout << (x == y); // returns 0 (false) because 5 is not equal to 3

   return 0;
}
```

```
0
Process returned 0 (0x0)   execution time : 0.292 s
Press any key to continue.
```

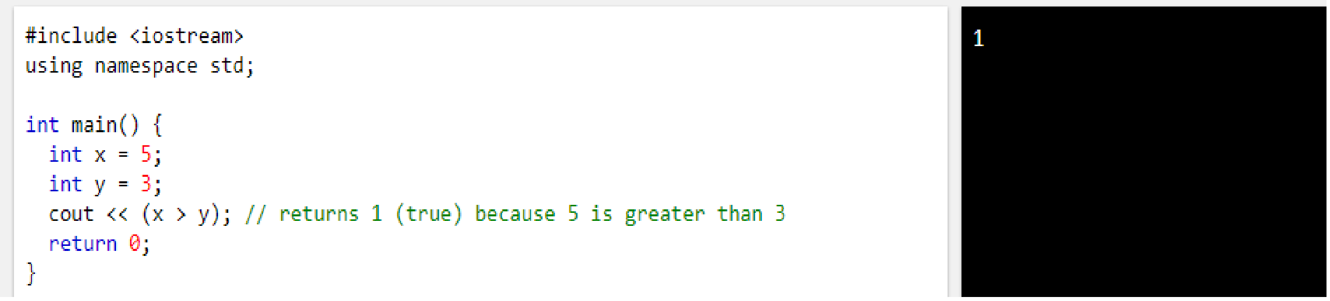
Example2:



```cpp
#include <iostream>

using namespace std;

int main()
{
  int x = 5;
  int y = 3;
  cout << (x != y); // returns 0 (false) because 5 is not equal to 3

   return 0;
}
```

```
1
Process returned 0 (0x0)   execution time : 1.004 s
Press any key to continue.
```

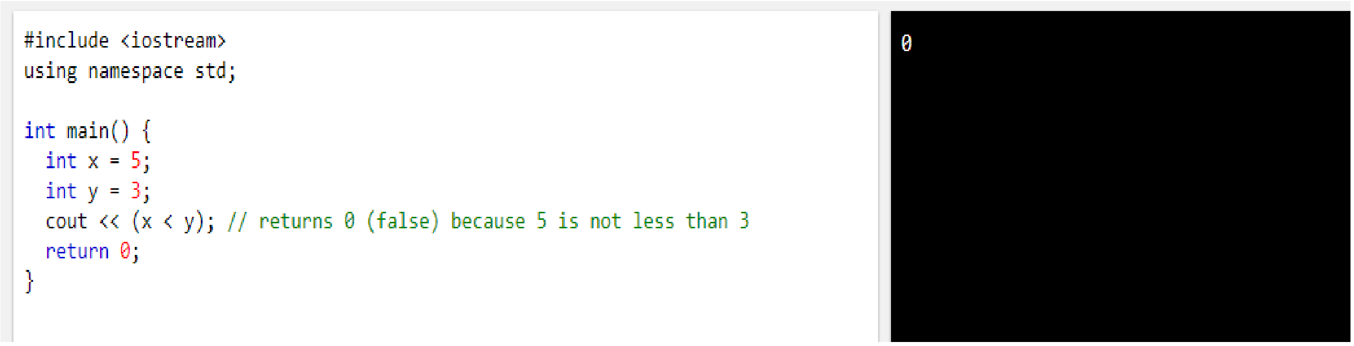
Example 3



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

int main() {
  int x = 5;
  int y = 3;
  cout << (x > y); // returns 1 (true) because 5 is greater than 3
  return 0;
}
```

```
1
```

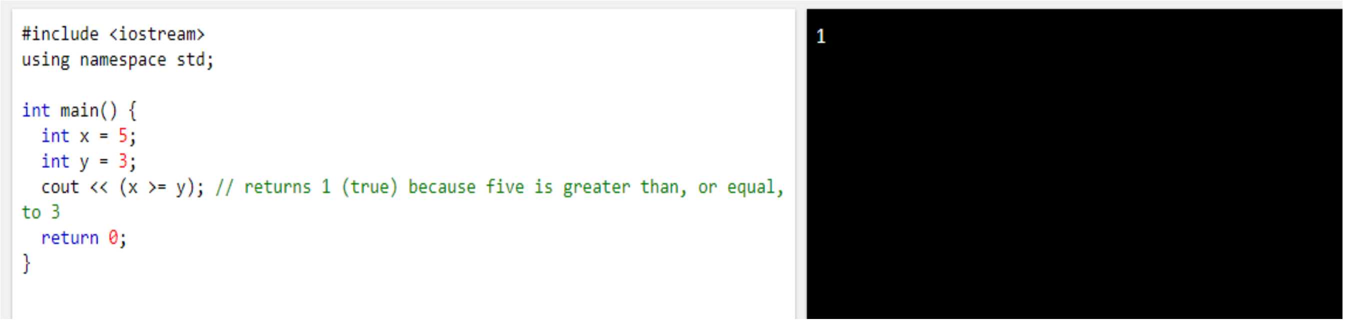
Example 4:



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

int main() {
  int x = 5;
  int y = 3;
  cout << (x < y); // returns 0 (false) because 5 is not less than 3
  return 0;
}
```

```
0
```

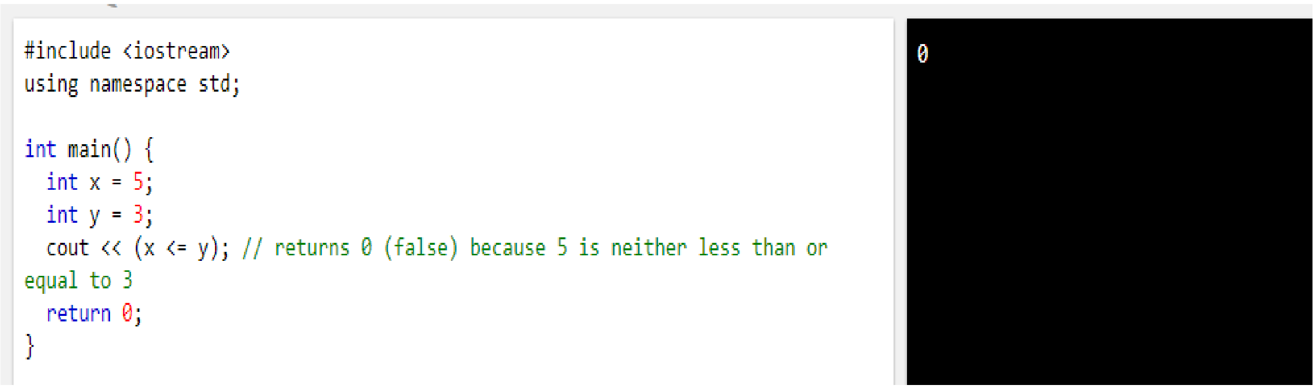
Example 5:



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

int main() {
  int x = 5;
  int y = 3;
  cout << (x >= y); // returns 1 (true) because five is greater than, or equal,
to 3
  return 0;
}
```

```
1
```

Example 6

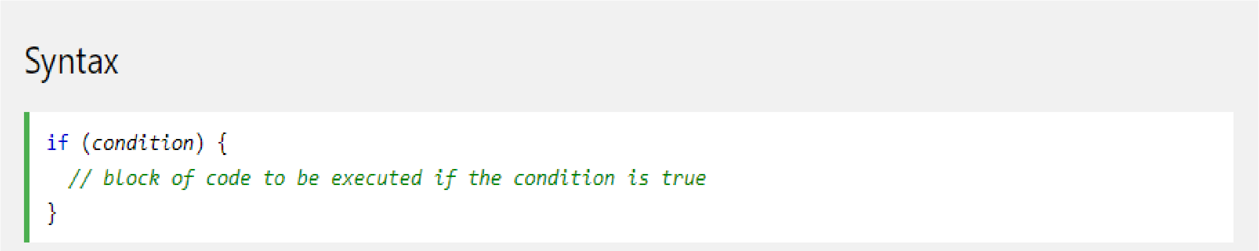


```
#include <iostream>
using namespace std;

int main() {
  int x = 5;
  int y = 3;
  cout << (x <= y); // returns 0 (false) because 5 is neither less than or
equal to 3
  return 0;
}
```

```
0
```

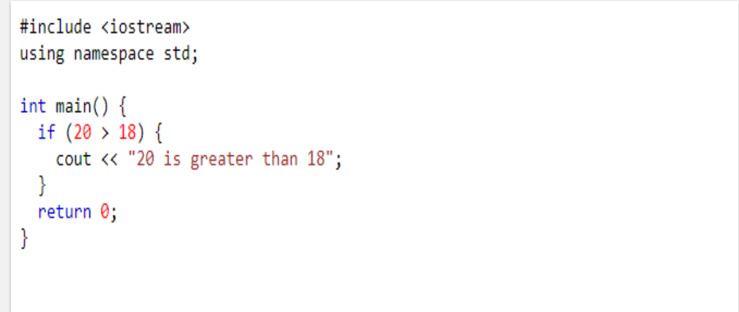
If Statements

if to specify a block of code to be executed, if a specified condition is true.

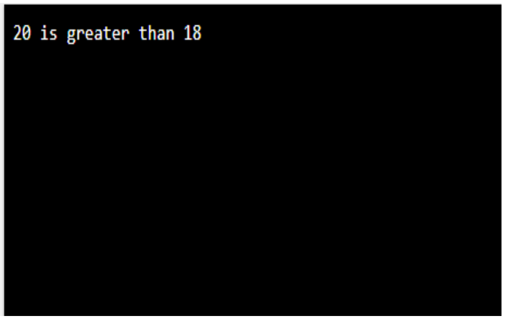


Syntax

```
if (condition) {
  // block of code to be executed if the condition is true
}
```

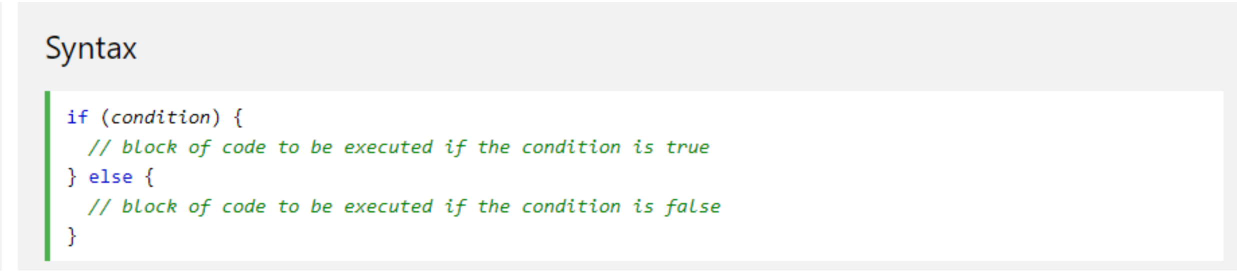


```
#include <iostream>
using namespace std;

int main() {
  if (20 > 18) {
    cout << "20 is greater than 18";
  }
  return 0;
}
```



```
20 is greater than 18
```

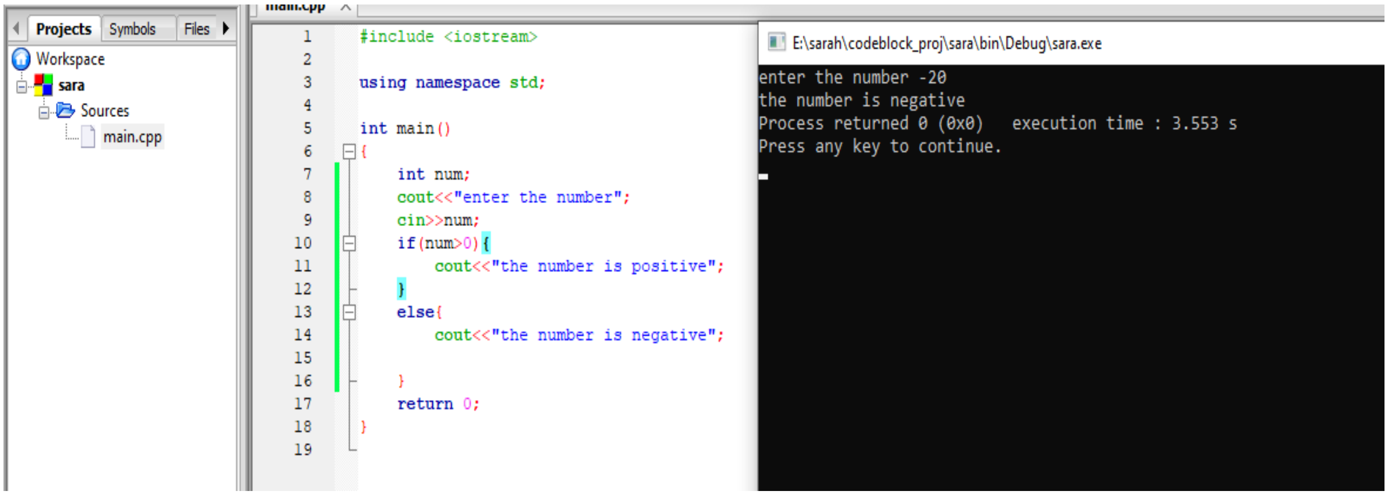
The else Statement



Syntax

```
if (condition) {
  // block of code to be executed if the condition is true
} else {
  // block of code to be executed if the condition is false
}
```

Write a c++ program to check the number negative or positive?



```
#include <iostream>

using namespace std;

int main()
{
    int num;
    cout<<"enter the number";
    cin>>num;
    if(num>0){
        cout<<"the number is positive";
    }
    else{
        cout<<"the number is negative";

    }
    return 0;
}
```

```
enter the number -20
the number is negative
Process returned 0 (0x0)   execution time : 3.553 s
Press any key to continue.
```

Write a program to calculate hourly wages?

There are two choices

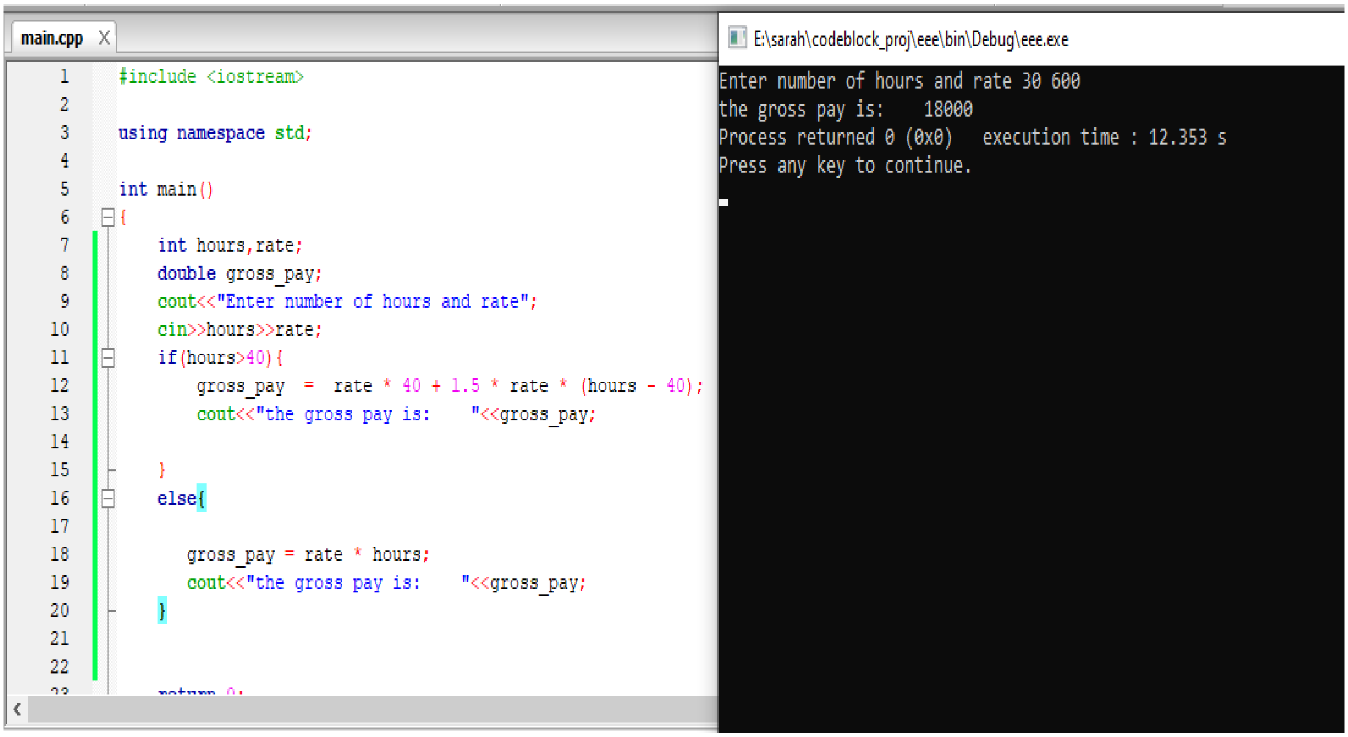
- Regular time (up to 40 hours)

  gross_pay = rate * hours;

- Overtime (over 40 hours)

  gross_pay = rate * 40 + 1.5 * rate * (hours - 40);

The program must choose which of these expressions to use

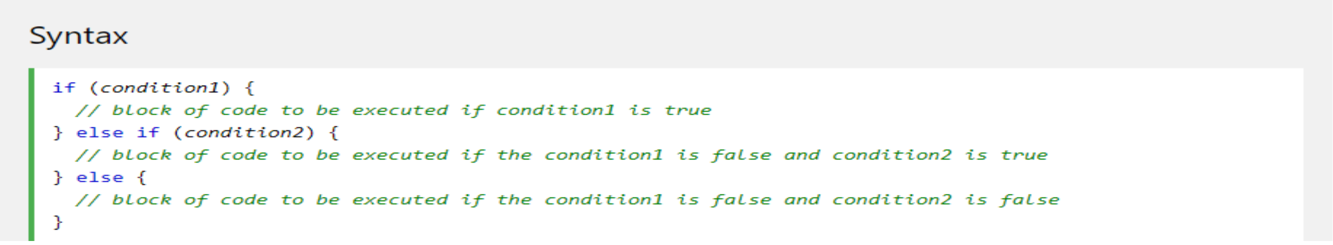


```cpp
#include <iostream>

using namespace std;

int main()
{
    int hours,rate;
    double gross_pay;
    cout<<"Enter number of hours and rate";
    cin>>hours>>rate;
    if(hours>40){
        gross_pay  =  rate * 40 + 1.5 * rate * (hours - 40);
        cout<<"the gross pay is:    "<<gross_pay;

    }
    else{

       gross_pay = rate * hours;
       cout<<"the gross pay is:    "<<gross_pay;
    }


    return 0;
```

```
Enter number of hours and rate 30 600
the gross pay is:    18000
Process returned 0 (0x0)   execution time : 12.353 s
Press any key to continue.
```

The else if Statement

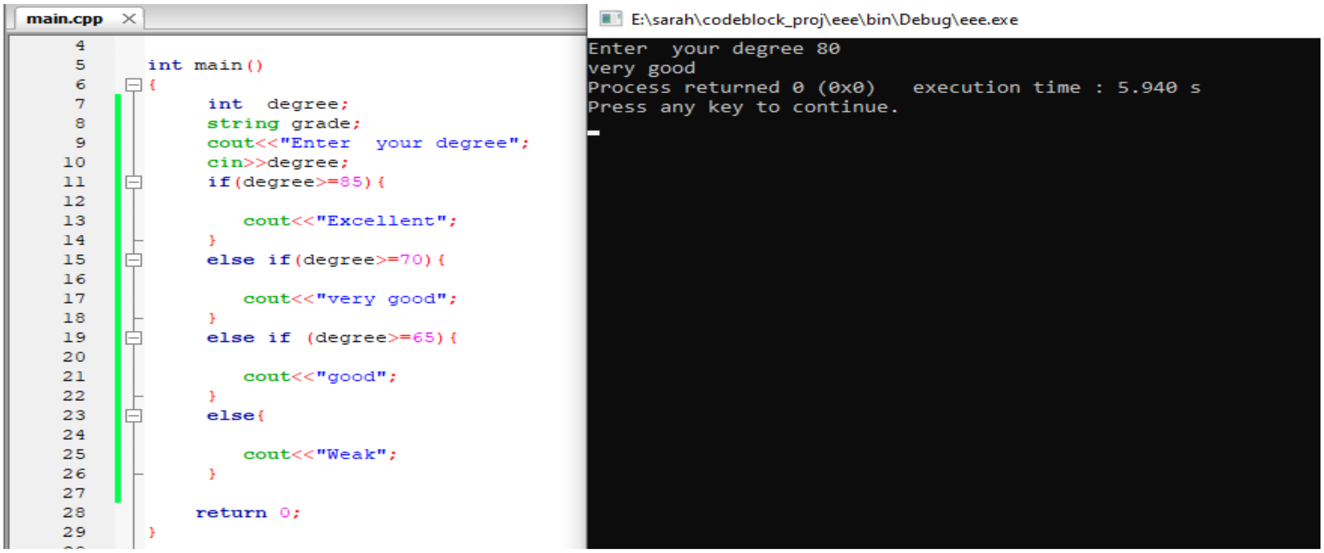
Use the else if statement to specify a new condition if the first condition is false



Syntax

```
if (condition1) {
  // block of code to be executed if condition1 is true
} else if (condition2) {
  // block of code to be executed if the condition1 is false and condition2 is true
} else {
  // block of code to be executed if the condition1 is false and condition2 is false
}
```

Example: write aprogram to Calculates students grades?



```
int main()
{
        int  degree;
        string grade;
        cout<<"Enter  your degree";
        cin>>degree;
        if(degree>=85){

           cout<<"Excellent";
        }
        else if(degree>=70){

           cout<<"very good";
        }
        else if  (degree>=65){

           cout<<"good";
        }
        else{

           cout<<"Weak";
        }

       return 0;
}
```

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

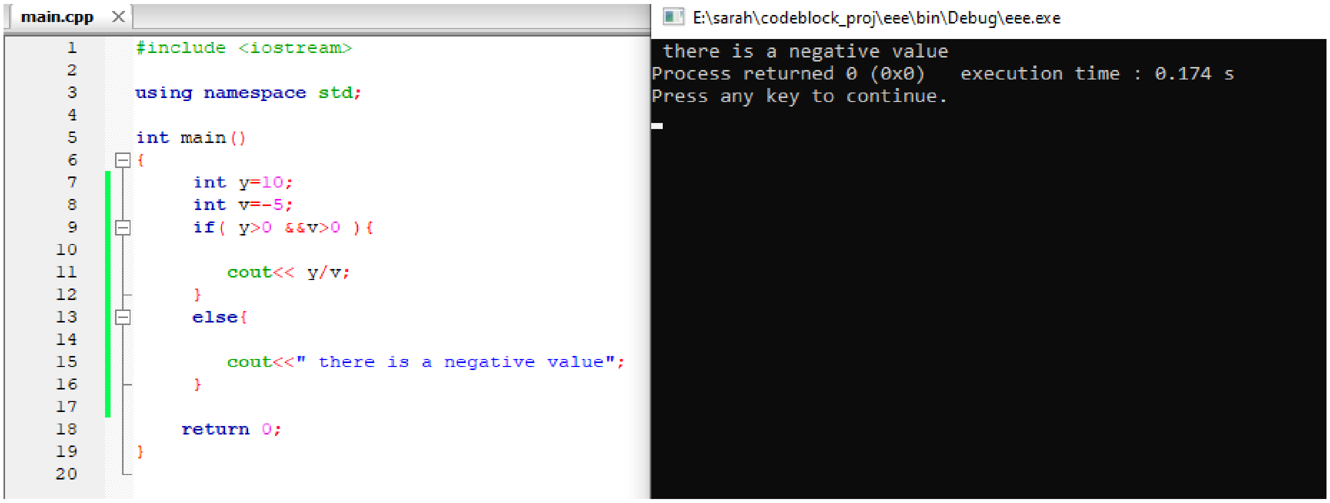
```
Enter  your degree 80
very good
Process returned 0 (0x0)   execution time : 5.940 s
Press any key to continue.
```

.

Logical Operators

Operator	Name	Description	Example
&&	Logical and	Returns true if both statements are true	x < 5 && x < 10
||	Logical or	Returns true if one of the statements is true	x < 5 || x < 4
!	Logical not	Reverse the result, returns false if the result is true	!(x < 5 && x < 10)

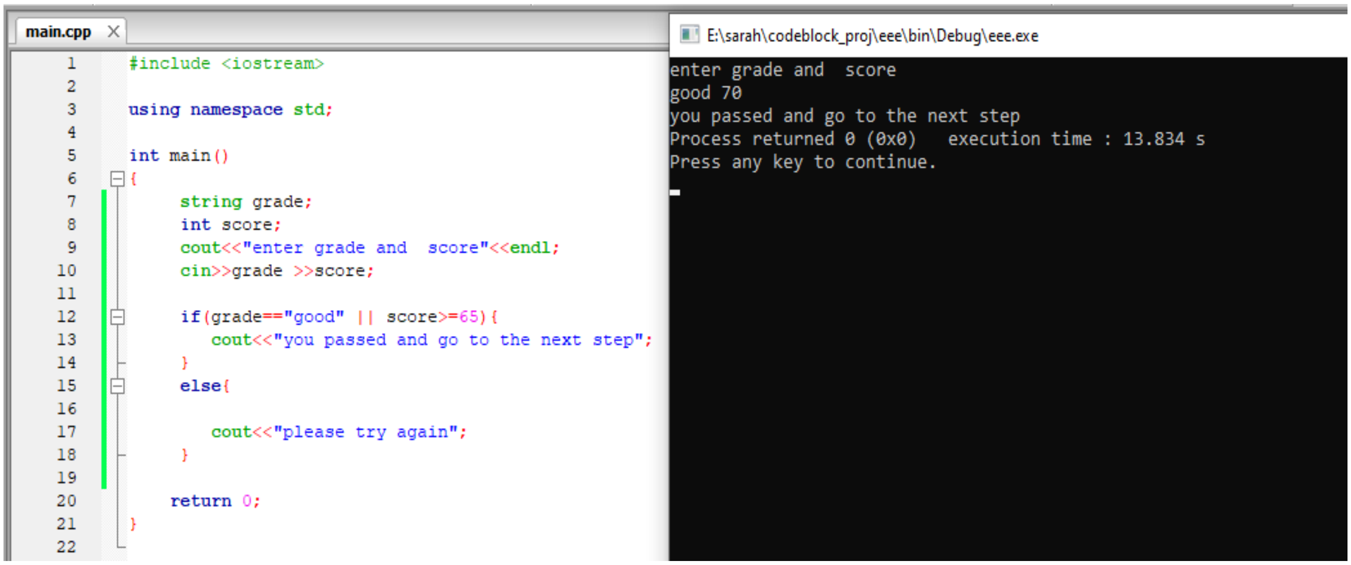
Example for And &&



```cpp
#include <iostream>

using namespace std;

int main()
{
     int y=10;
     int v=-5;
     if( y>0 &&v>0 ){

        cout<< y/v;
     }
     else{

        cout<<" there is a negative value";
     }

    return 0;
}

```

```
 there is a negative value
Process returned 0 (0x0)   execution time : 0.174 s
Press any key to continue.
```

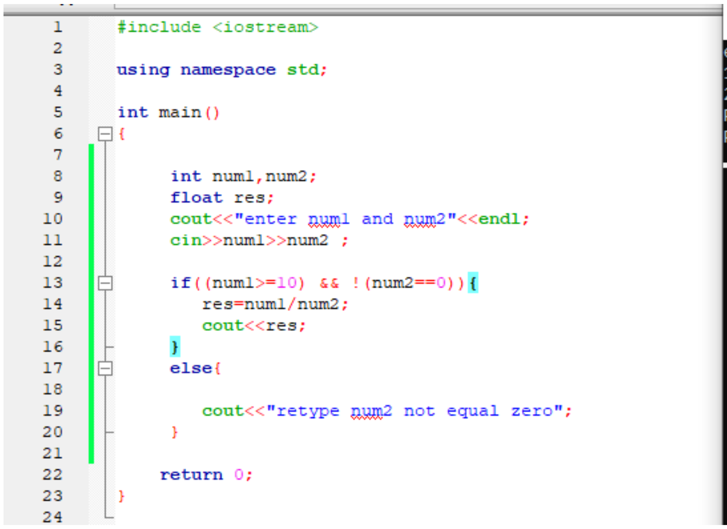
Example :Or ||



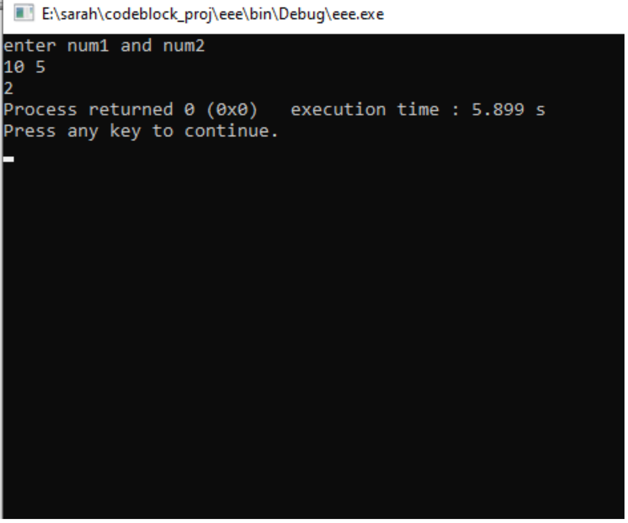
```cpp
#include <iostream>

using namespace std;

int main()
{
     string grade;
     int score;
     cout<<"enter grade and  score"<<endl;
     cin>>grade >>score;

     if(grade=="good" || score>=65){
        cout<<"you passed and go to the next step";
     }
     else{

        cout<<"please try again";
     }

    return 0;
}

```

```
enter grade and  score
good 70
you passed and go to the next step
Process returned 0 (0x0)   execution time : 13.834 s
Press any key to continue.
```

Example for Not



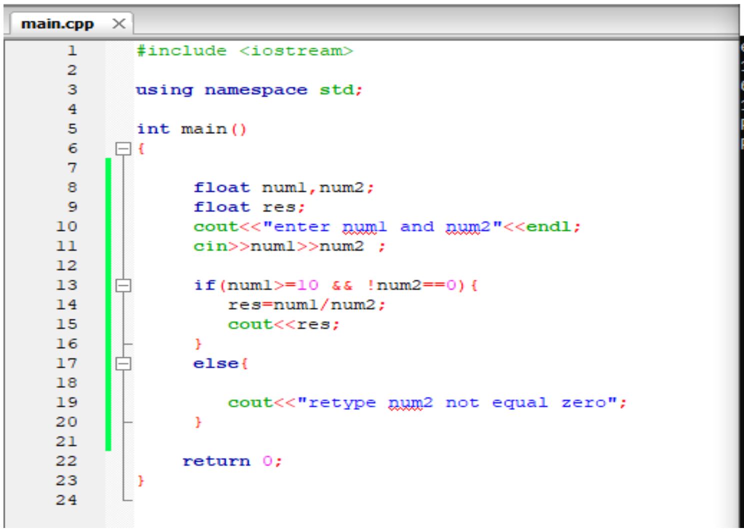
```
#include <iostream>

using namespace std;

int main()
{

    int num1,num2;
    float res;
    cout<<"enter num1 and num2"<<endl;
    cin>>num1>>num2 ;

    if((num1>=10) && !(num2==0)){
       res=num1/num2;
       cout<<res;
    }
    else{

       cout<<"retype num2 not equal zero";
    }

   return 0;
}
```



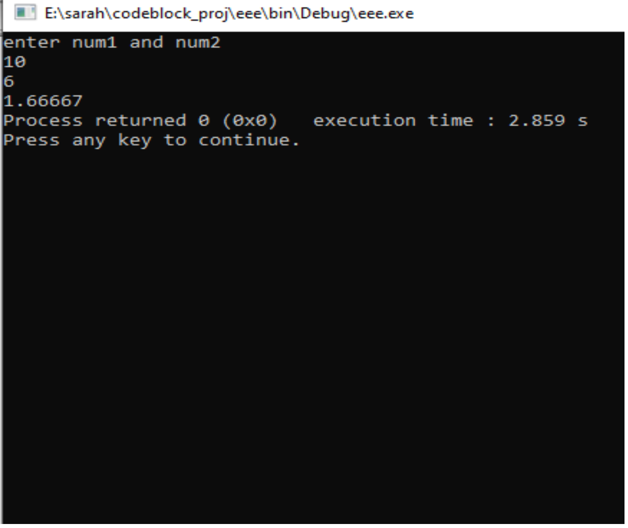
E:\sarah\codeblock_proj\eee\bin\Debug\eee.exe

```
enter num1 and num2
10 5
2
Process returned 0 (0x0)   execution time : 5.899 s
Press any key to continue.
```

This also right



```
#include <iostream>

using namespace std;

int main()
{

    float num1,num2;
    float res;
    cout<<"enter num1 and num2"<<endl;
    cin>>num1>>num2 ;

    if(num1>=10 && !num2==0){
       res=num1/num2;
       cout<<res;
    }
    else{

       cout<<"retype num2 not equal zero";
    }

   return 0;
}
```



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

```
enter num1 and num2
10
6
1.66667
Process returned 0 (0x0)   execution time : 2.859 s
Press any key to continue.
```

C++ Switch Statements

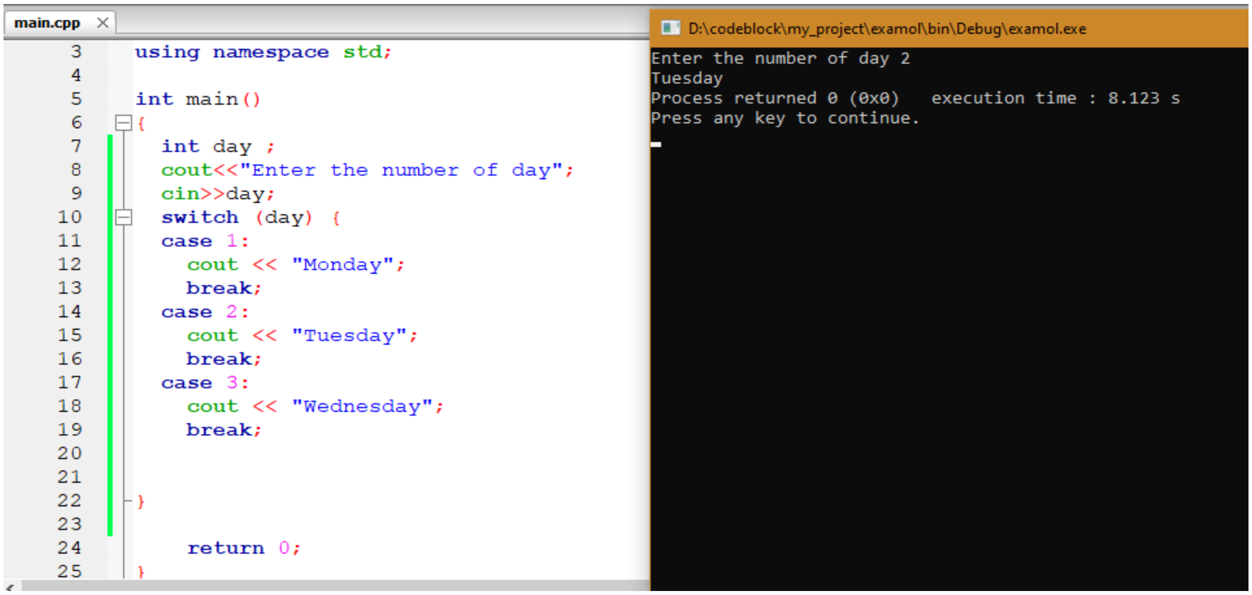
Use the `switch` statement to select one of many code blocks to be executed.

Syntax

```
switch(expression) {
  case x:
    // code block
    break;
  case y:
    // code block
    break;
  default:
    // code block
}
```

This is how it works:

- The `switch` expression is evaluated once
- The value of the expression is compared with the values of each `case`
- If there is a match, the associated block of code is executed
- The `break` and `default` keywords are optional, and will be described later in this chapter

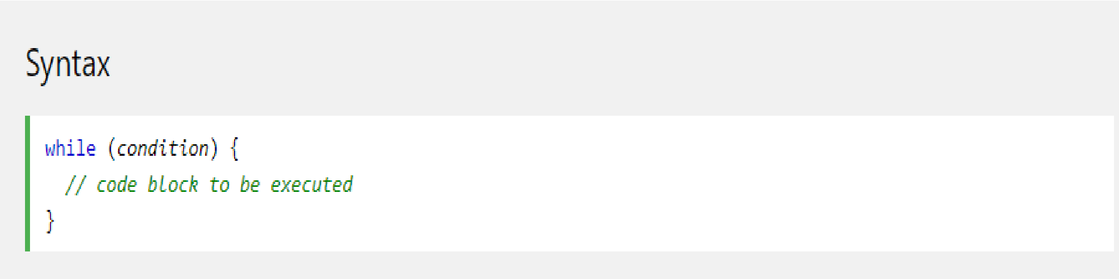


```
using namespace std;

int main()
{
  int day ;
  cout<<"Enter the number of day";
  cin>>day;
  switch (day) {
  case 1:
    cout << "Monday";
    break;
  case 2:
    cout << "Tuesday";
    break;
  case 3:
    cout << "Wednesday";
    break;


}

    return 0;
}
```

```
Enter the number of day 2
Tuesday
Process returned 0 (0x0)   execution time : 8.123 s
Press any key to continue.
```

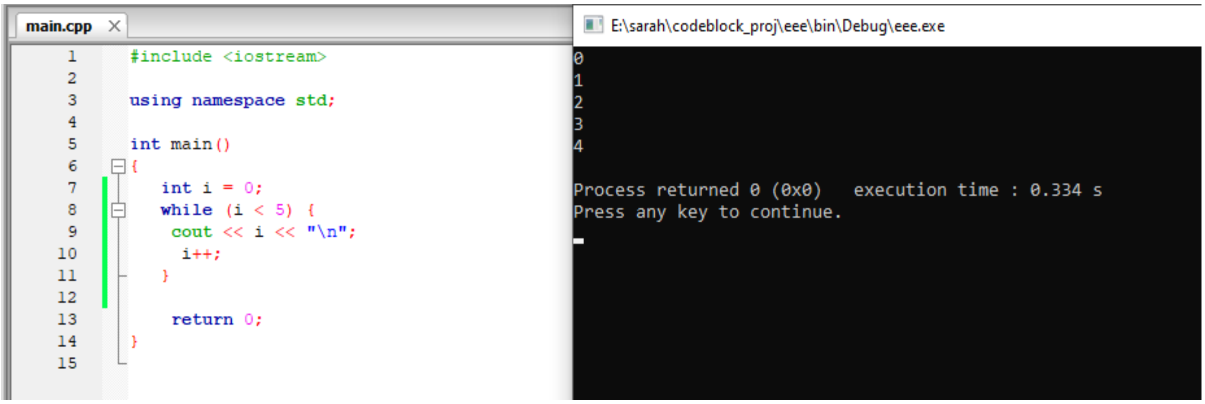
While Loop Operation

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
}
```

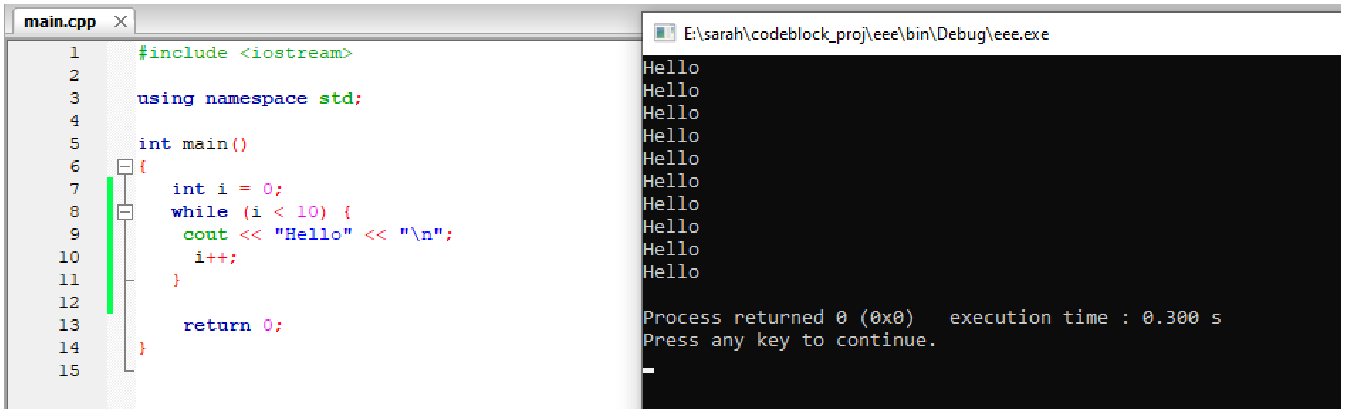


```cpp
#include <iostream>

using namespace std;

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

    return 0;
}
```

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

```
0
1
2
3
4

Process returned 0 (0x0)   execution time : 0.334 s
Press any key to continue.
```

Write a c++ program print Hello 10 times?

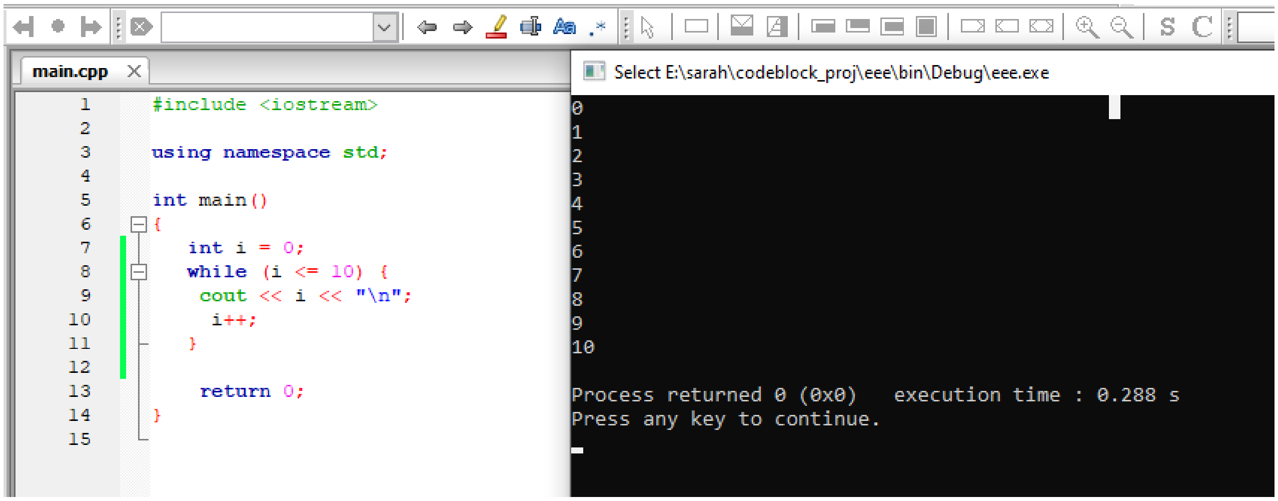


```cpp
#include <iostream>

using namespace std;

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

    return 0;
}
```

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

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

Process returned 0 (0x0)   execution time : 0.300 s
Press any key to continue.
```

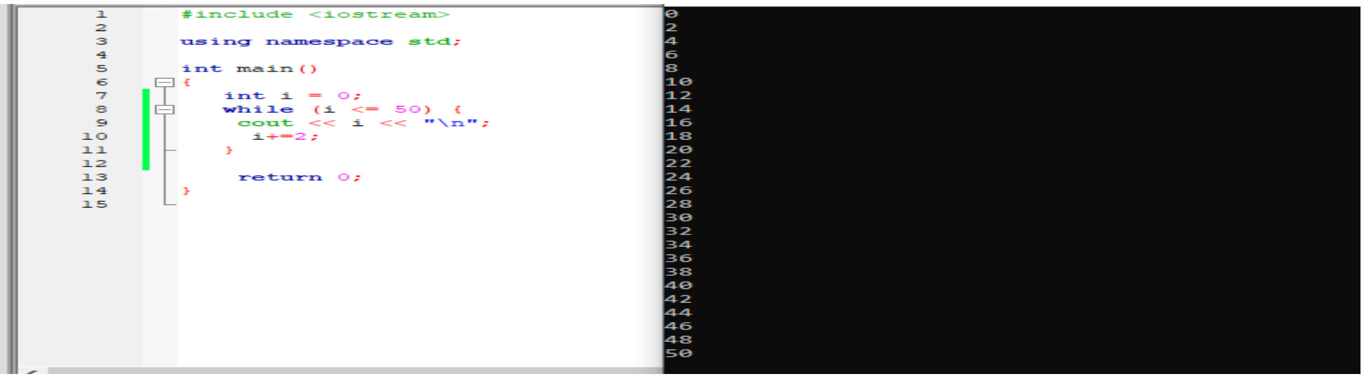
Write a c++ program to print number from 0 to 10?



```cpp
#include <iostream>

using namespace std;

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

    return 0;
}
```

```
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.
```

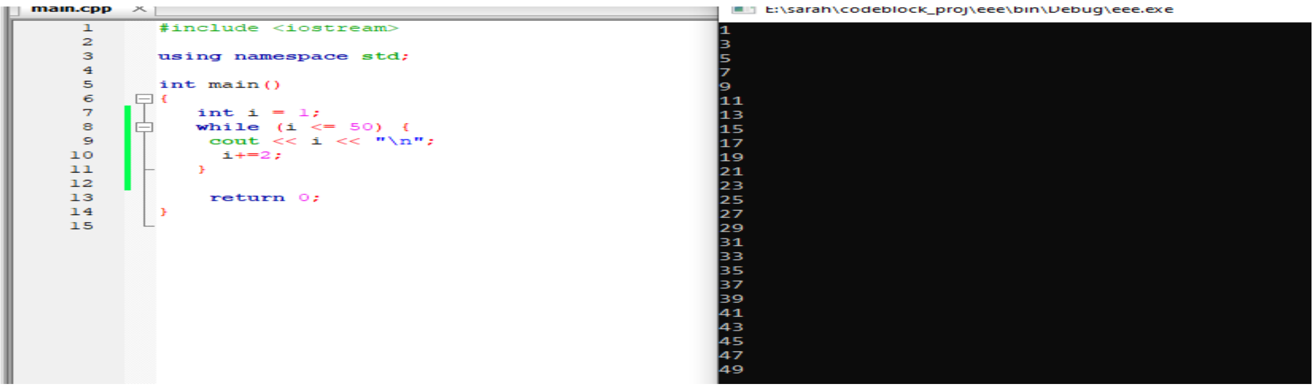
Write a c++ program to print even numbers until 50?



```cpp
#include <iostream>

using namespace std;

int main()
{
   int i = 0;
   while (i <= 50) {
    cout << i << "\n";
     i+=2;
   }

    return 0;
}
```

```
0
2
4
6
8
10
12
14
16
18
20
22
24
26
28
30
32
34
36
38
40
42
44
46
48
50
```

Write a c++ program to print odd numbers until 50?

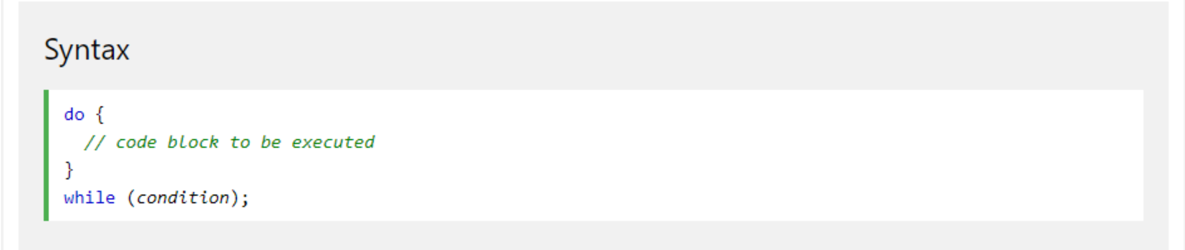


```cpp
#include <iostream>

using namespace std;

int main()
{
   int i = 1;
   while (i <= 50) {
    cout << i << "\n";
     i+=2;
   }

    return 0;
}
```

```
1
3
5
7
9
11
13
15
17
19
21
23
25
27
29
31
33
35
37
39
41
43
45
47
49
```

do-while loop

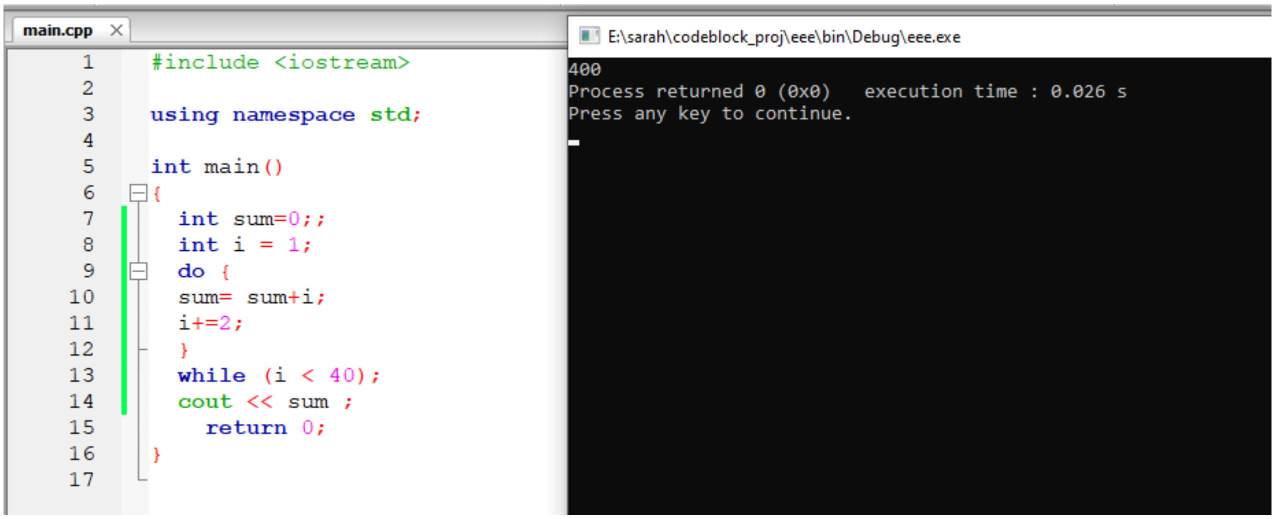
The do/while loop is a variant of the while loop. This loop will execute the code block once, before checking if the condition is true, then it will repeat the loop as long as the condition is true.



Syntax

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

Write aprogram to print sum of odd numbers <40?



```
#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)   execution time : 0.026 s
Press any key to continue.
```

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

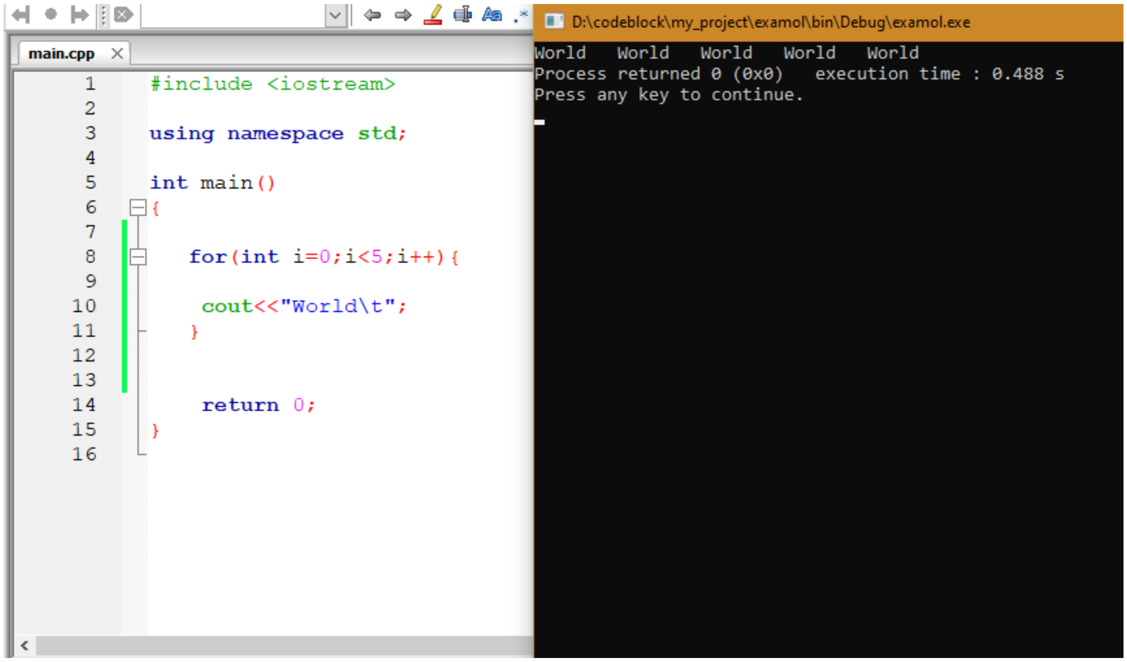
```
for (statement 1; statement 2; statement 3) {
  // code block to be executed
}
```

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.

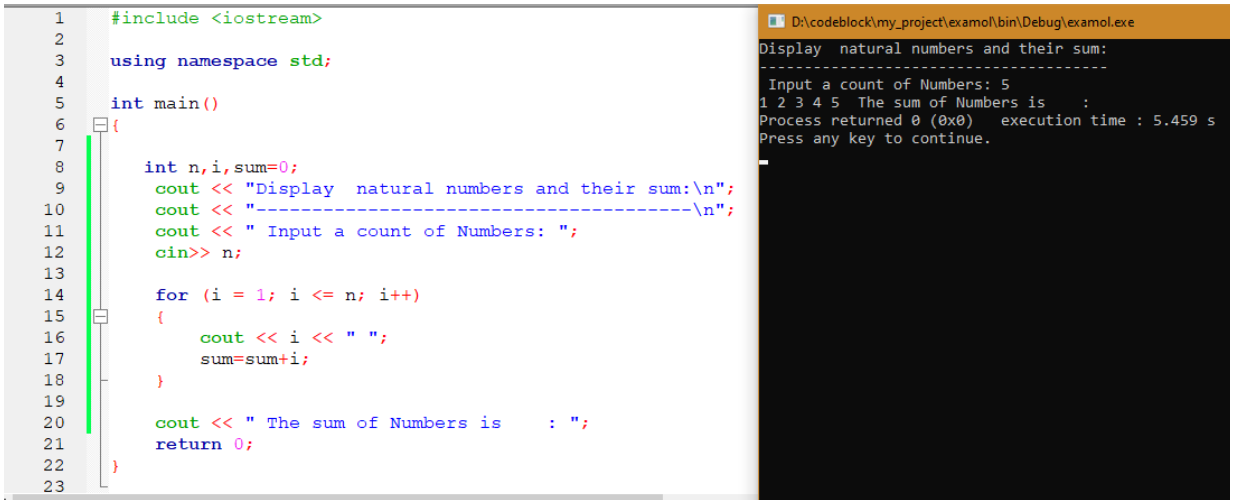
Write a c++ program to print world 4 times?



```
#include <iostream>

using namespace std;

int main()
{

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

    cout<<"World\t";
   }


    return 0;
}
```

```
World   World   World   World   World
Process returned 0 (0x0)   execution time : 0.488 s
Press any key to continue.
```

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

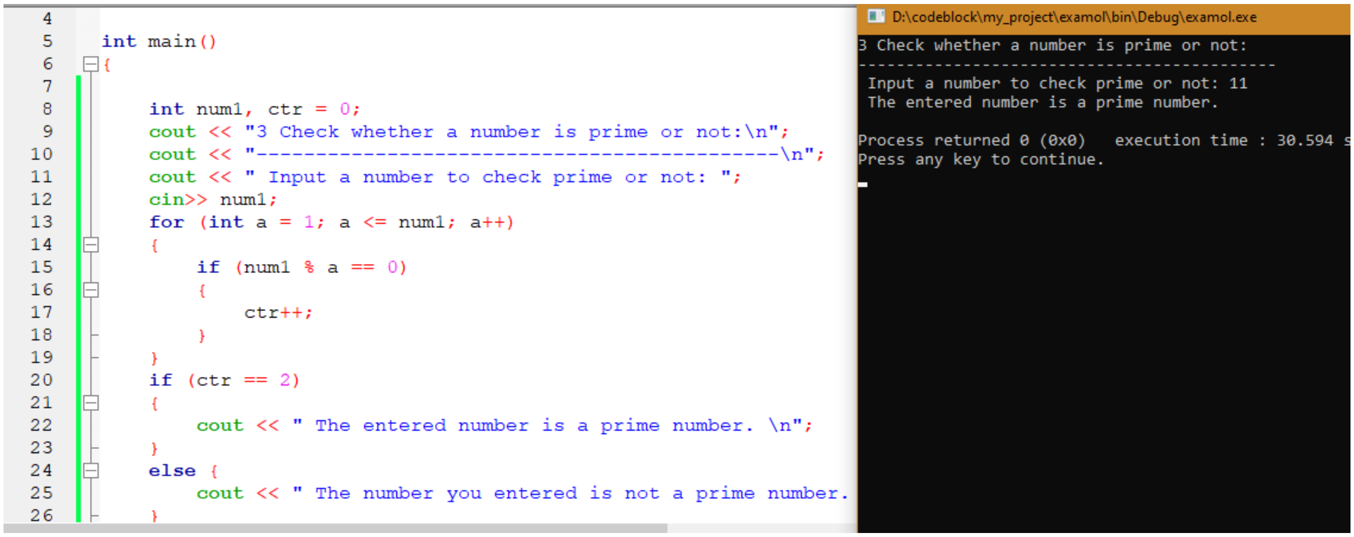


```cpp
#include <iostream>

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;
}

```

D:\codeblock\my_project\examol\bin\Debug\examol.exe

```
Display  natural numbers and their sum:
---------------------------------------
 Input a count of Numbers: 5
1 2 3 4 5  The sum of Numbers is     :
Process returned 0 (0x0)   execution time : 5.459 s
Press any key to continue.
```

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

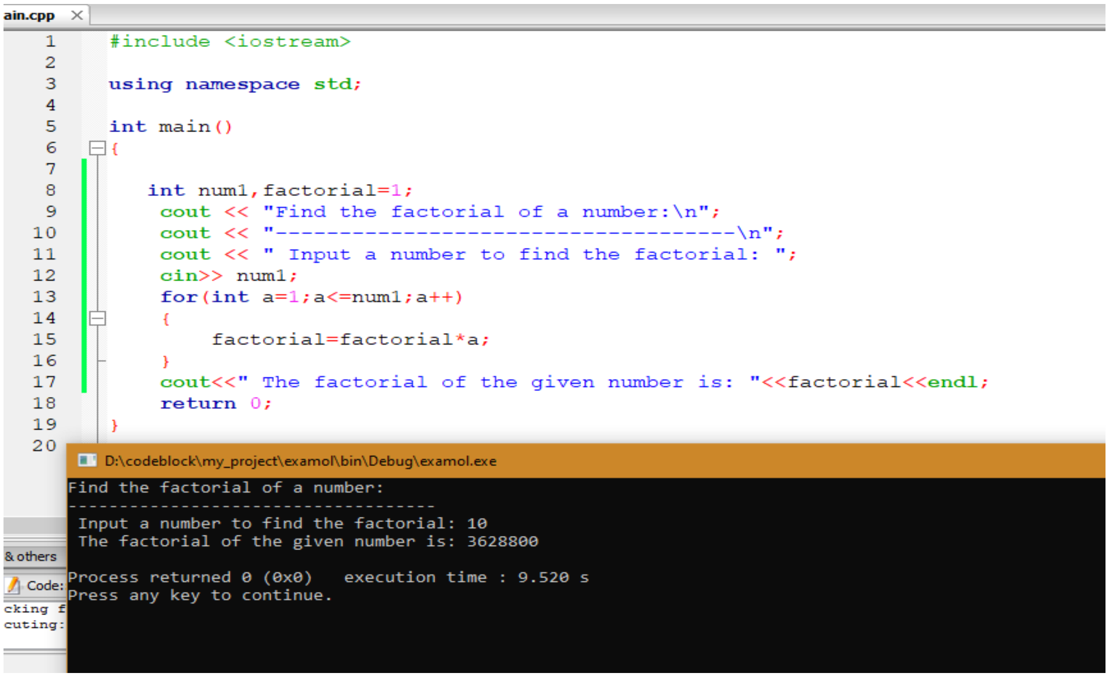


```cpp

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 : 30.594 s
Press any key to continue.
```

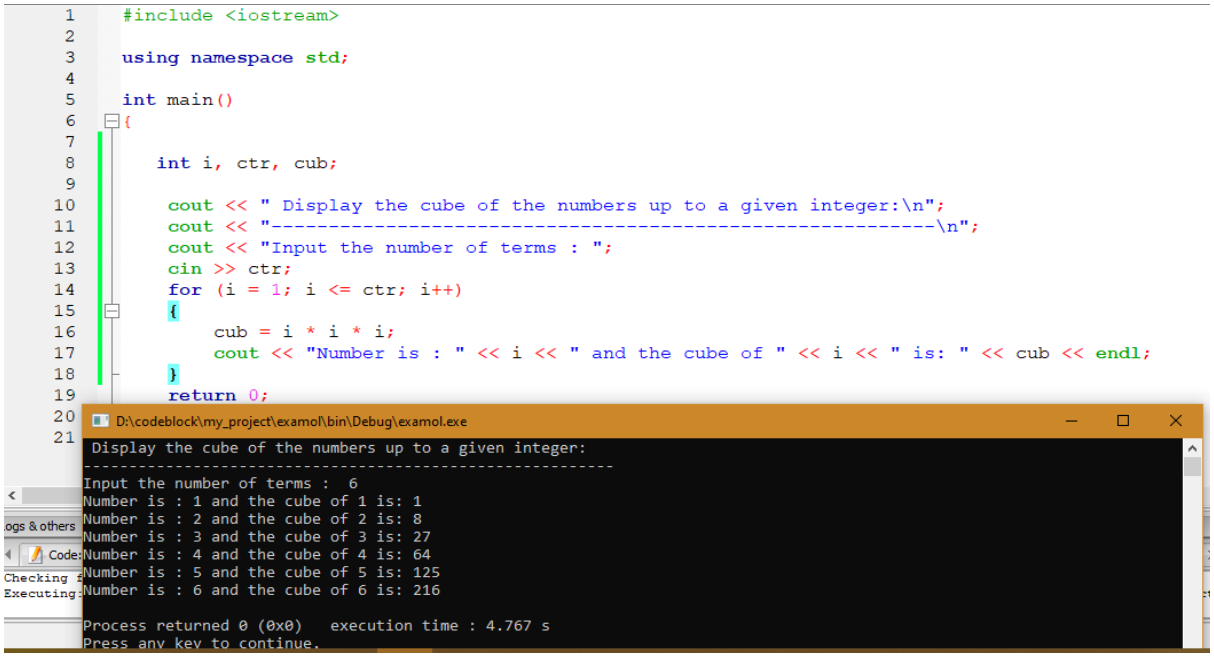
Write a program in C++ to find the factorial of a number?



```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;
}
```

```
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.
```

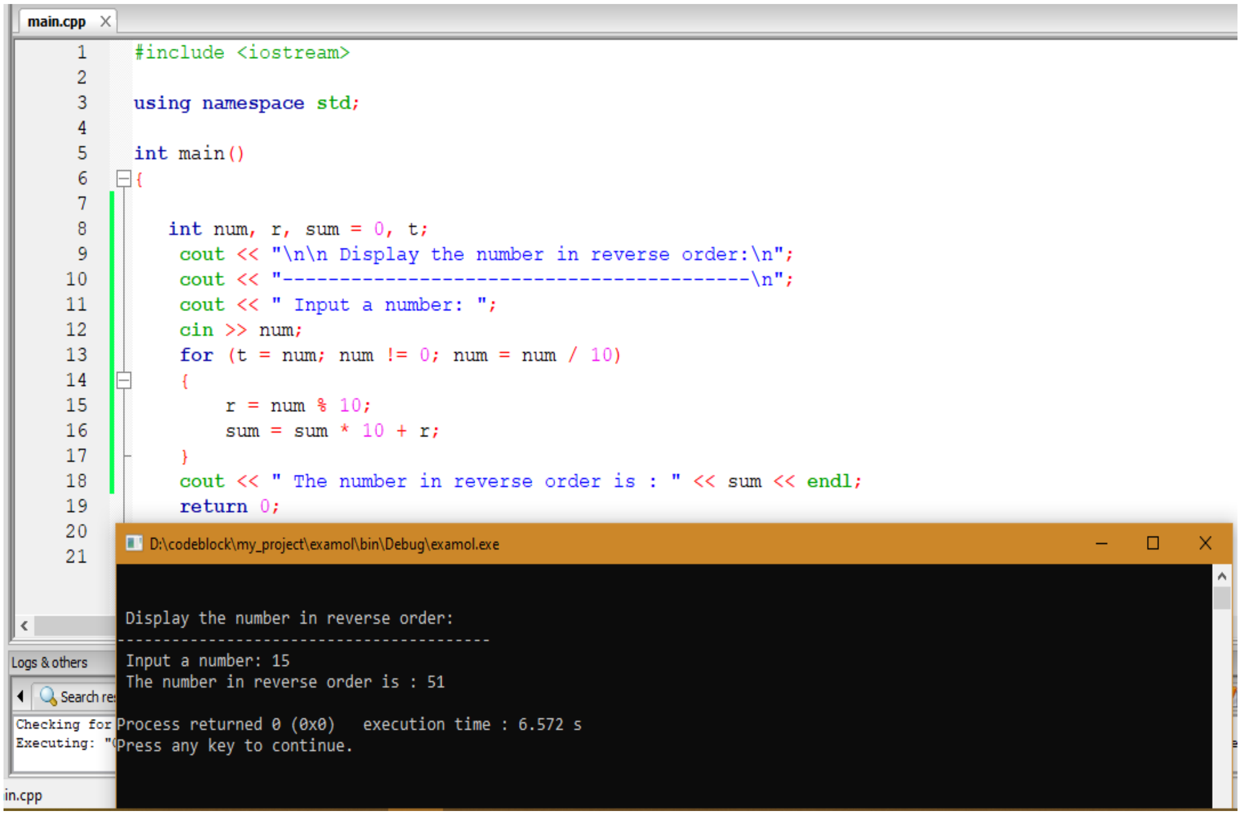
Write a program in C++ to display the cube of the number up to given an integer?



```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;
```

```
 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

Process returned 0 (0x0)   execution time : 4.767 s
Press any key to continue.
```

Write a program in C++ to display the number in reverse order?



```cpp
#include <iostream>

using namespace std;

int main()
{

    int num, r, sum = 0, t;
     cout << "\n\n Display the number in reverse order:\n";
     cout << "----------------------------------------\n";
     cout << " Input a number: ";
     cin >> num;
     for (t = num; num != 0; num = num / 10)
     {
         r = num % 10;
         sum = sum * 10 + r;
     }
     cout << " The number in reverse order is : " << sum << endl;
     return 0;
```

```


 Display the number in reverse order:
----------------------------------------
 Input a number: 15
 The number in reverse order is : 51

Process returned 0 (0x0)   execution time : 6.572 s
Press any key to continue.
```

Tasks:

- Write an if-else statement that outputs the word **High** if the value of the variable score is greater than 100 and **Low** if the value of score is at most 100? The variables are of type int?

- Write an if-else statement that outputs the word **Warning** provided that either the value of the variable temperature is greater than or equal to 100, or the of the variable pressure is greater than or equal to 200, or both?

- Write a program when the color of traffic is green, print go, in the case of the traffic is yellow print ready, and if the traffic color is red, prints stop؟

- Write aprogram which print your name 10 times on screen؟

- Write a program to Solve the following formula: $\frac{y-c}{d+v}$ where y=10 , d=20 , Enter the c and v values during the program execution?

- Write a program in C++ to find the number and sum of all integer between 100 and 200 which are divisible by 9?