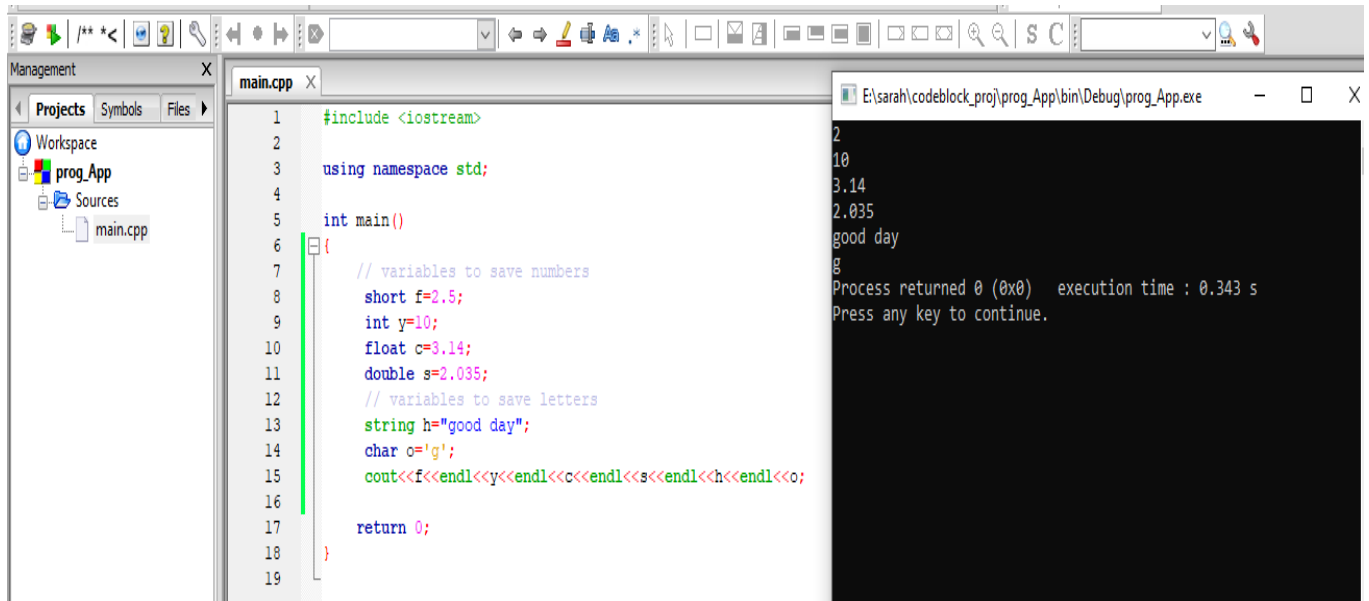


Exercises – chapter2

Variables and Assignments

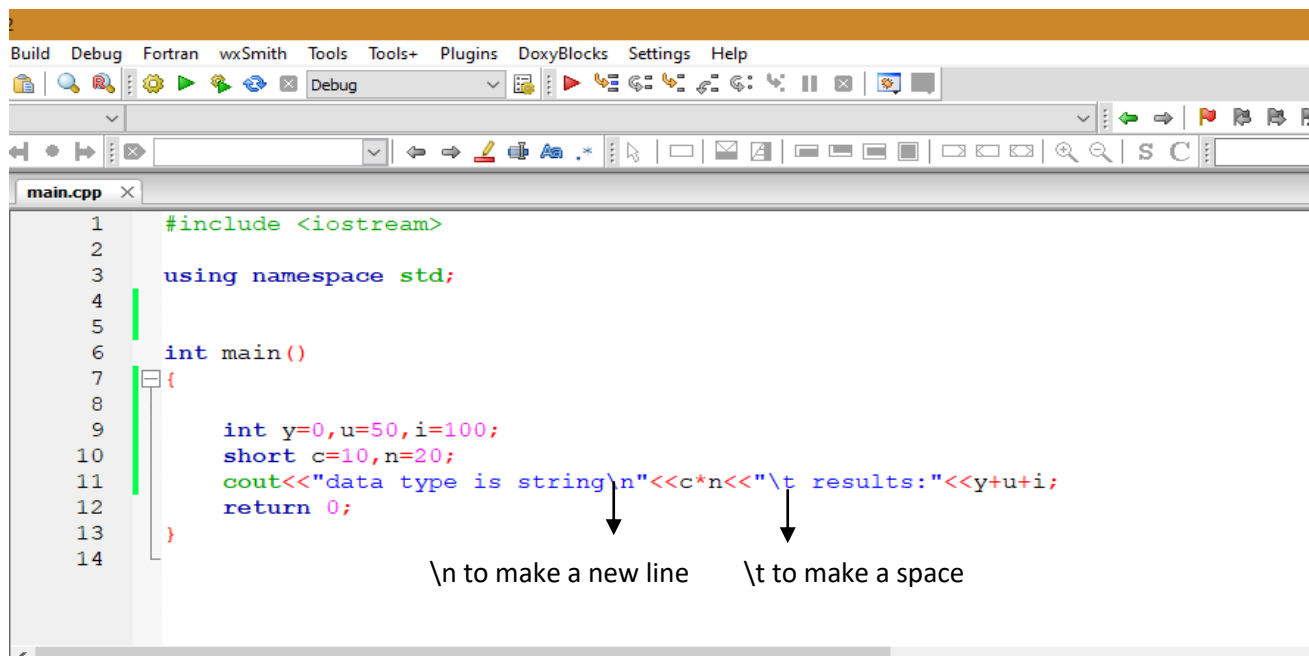


```
1 #include <iostream>
2
3 using namespace std;
4
5 int main()
6 {
7     // variables to save numbers
8     short f=2.5;
9     int y=10;
10    float c=3.14;
11    double s=2.035;
12    // variables to save letters
13    string h="good day";
14    char o='g';
15    cout<<f<<endl<<y<<endl<<c<<endl<<s<<endl<<h<<endl<<o;
16
17    return 0;
18 }
19
```

Terminal output:

```
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



```
1 #include <iostream>
2
3 using namespace std;
4
5
6 int main()
7 {
8
9     int y=0,u=50,i=100;
10    short c=10,n=20;
11    cout<<"data type is string\n"<<c*n<<"\t" results:"<<y+u+i;
12    return 0;
13 }
14
```

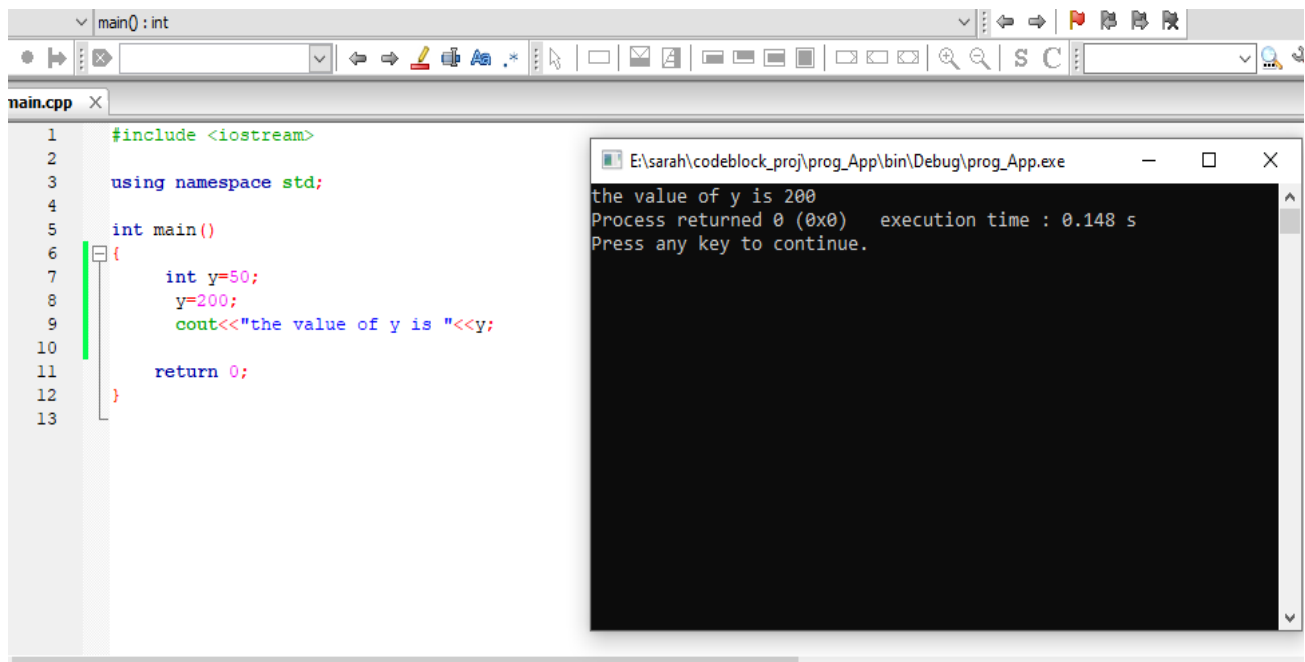
\n to make a new line \t to make a space

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:

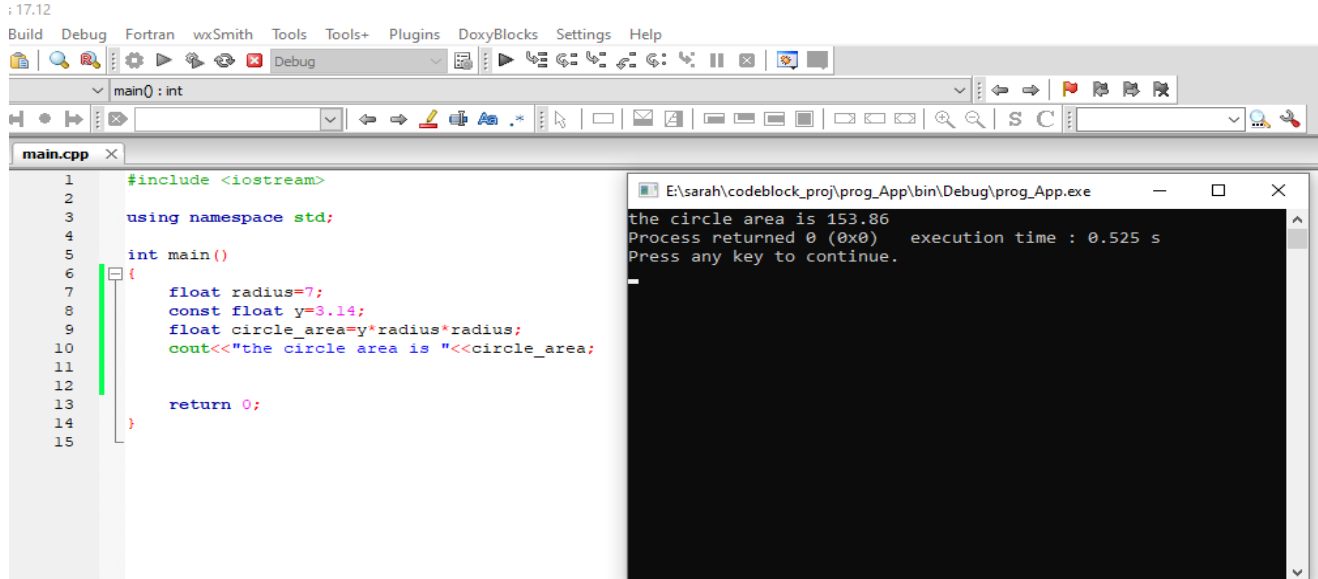


Constants

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

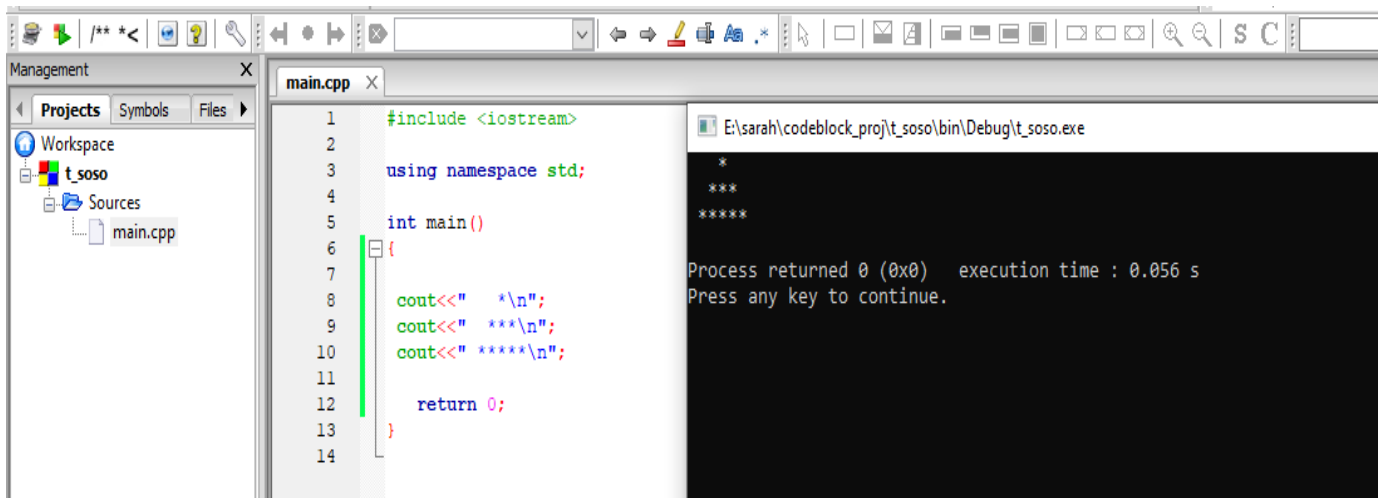
- ▶ Must declare and initialize constant in same time.
- ▶ Constant can't reinitialize.

Example: Compute circle area

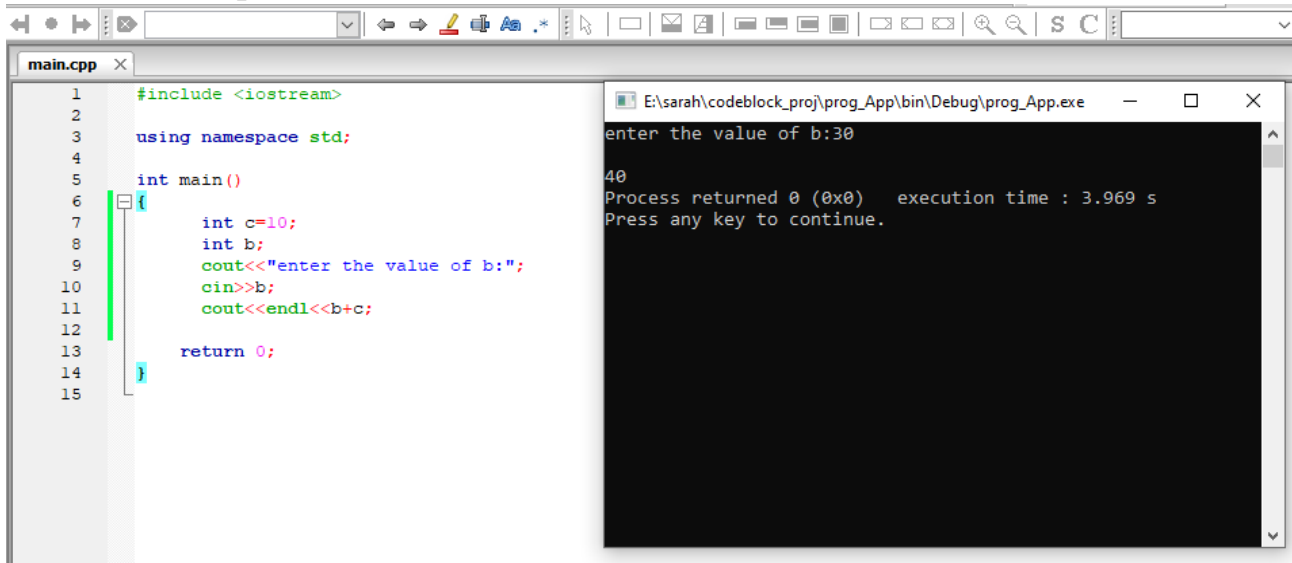


Input and Output

Example for cout command



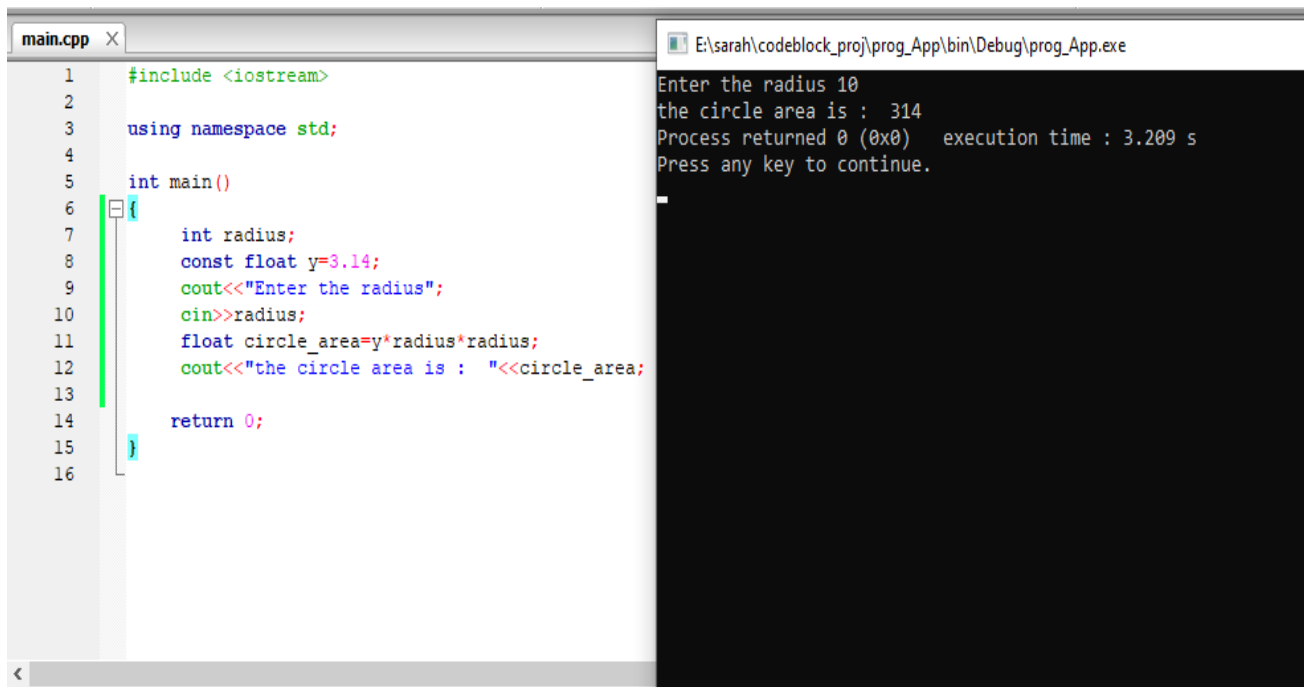
► cin is a predefined variable that reads data from the keyboard with the extraction operator (>>).



```
main.cpp x
1 #include <iostream>
2
3 using namespace std;
4
5 int main()
6 {
7     int c=10;
8     int b;
9     cout<<"enter the value of b:";
10    cin>>b;
11    cout<<endl<<b+c;
12
13    return 0;
14 }
15
```

```
E:\sarah\codeblock_proj\prog_App\bin\Debug\prog_App.exe
enter the value of b:30
40
Process returned 0 (0x0)   execution time : 3.969 s
Press any key to continue.
```

Example 2:Compute circle area by cin command

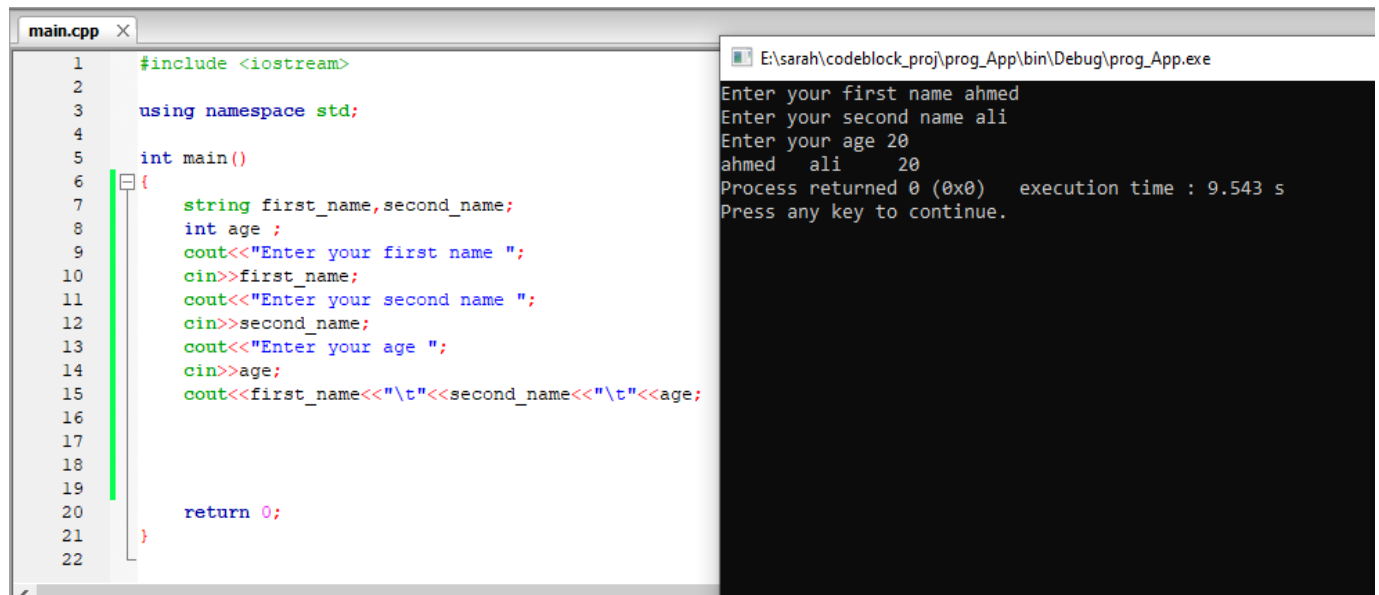


```
main.cpp x
1 #include <iostream>
2
3 using namespace std;
4
5 int main()
6 {
7     int radius;
8     const float y=3.14;
9     cout<<"Enter the radius";
10    cin>>radius;
11    float circle_area=y*radius*radius;
12    cout<<"the circle area is : "<<circle_area;
13
14    return 0;
15 }
16
```

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

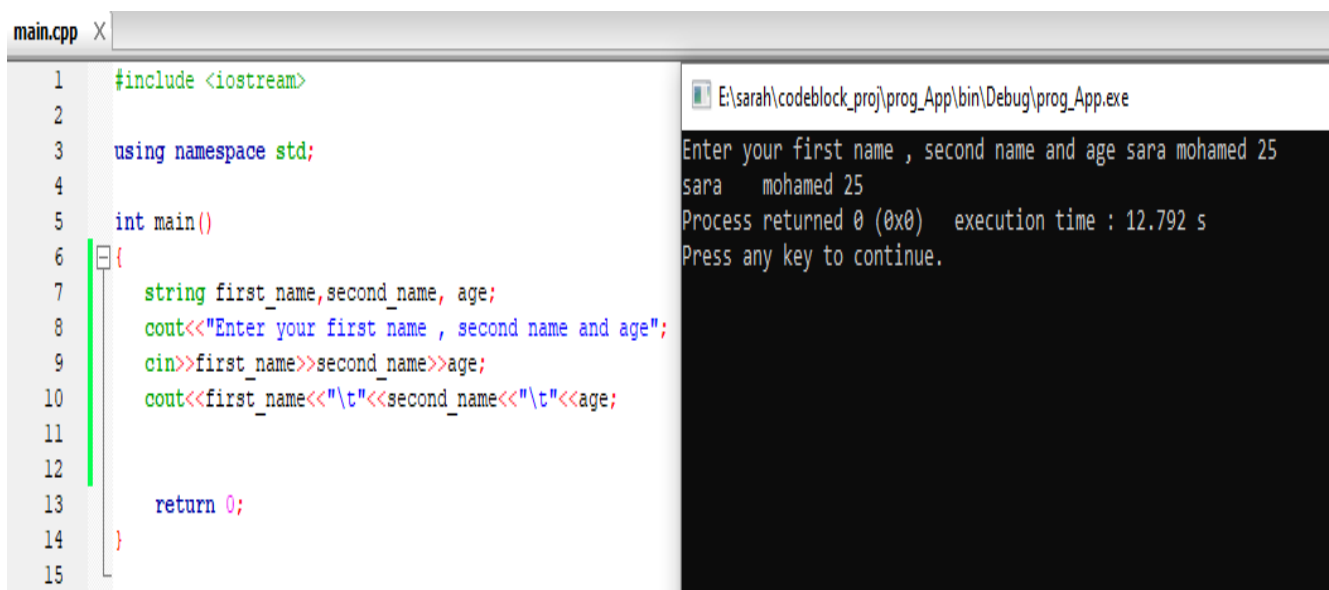
1. First method



```
main.cpp X
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      string first_name,second_name;
8      int age ;
9      cout<<"Enter your first name ";
10     cin>>first_name;
11     cout<<"Enter your second name ";
12     cin>>second_name;
13     cout<<"Enter your age ";
14     cin>>age;
15     cout<<first_name<<"\t"<<second_name<<"\t"<<age;
16
17
18
19
20     return 0;
21 }
22
```

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



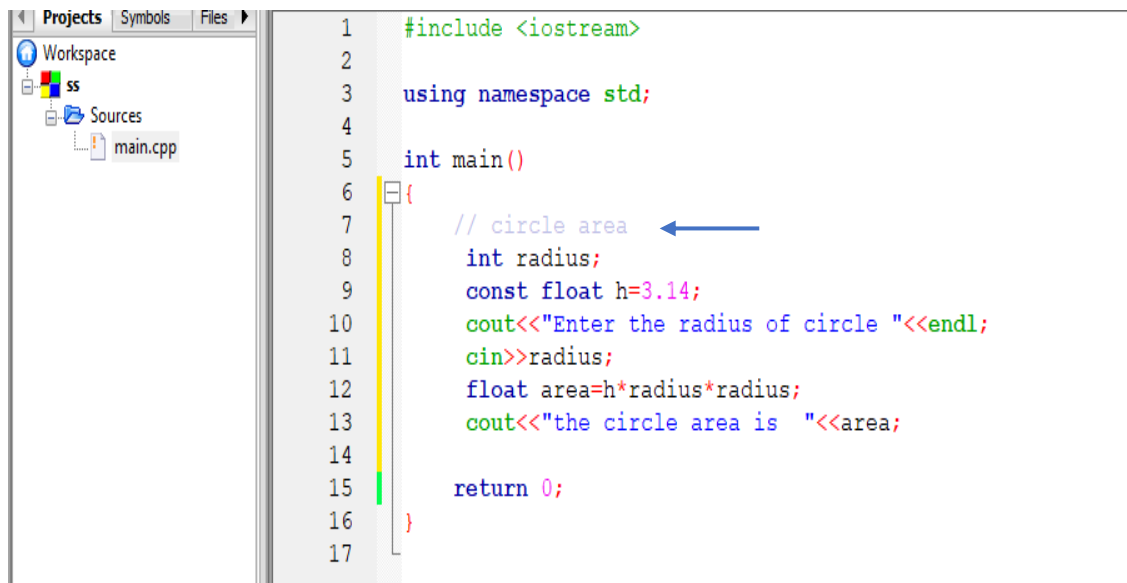
```
main.cpp X
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      string first_name,second_name, age;
8      cout<<"Enter your first name , second name and age";
9      cin>>first_name>>second_name>>age;
10     cout<<first_name<<"\t"<<second_name<<"\t"<<age;
11
12
13     return 0;
14 }
15
```

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

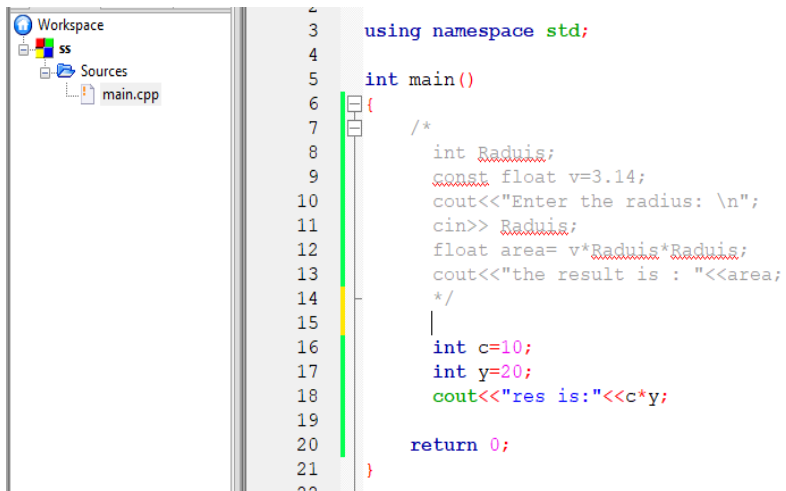
- Comments can be used to explain C++ code
- to make it more readable. It can also be used to prevent execution when testing alternative code.
- Comments can be **singled-lined** or **multi-lined**

Example for singled-lined comment



```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      // circle area ←
8      int radius;
9      const float h=3.14;
10     cout<<"Enter the radius of circle "<<endl;
11     cin>>radius;
12     float area=h*radius*radius;
13     cout<<"the circle area is  "<<area;
14
15     return 0;
16 }
17
```

Example for multi-lined comment



```
2
3  using namespace std;
4
5  int main()
6  {
7      /*
8       int Raduis;
9       const float v=3.14;
10      cout<<"Enter the radius: \n";
11      cin>> Raduis;
12      float area= v*Raduis*Raduis;
13      cout<<"the result is : "<<area;
14      */
15
16     int c=10;
17     int y=20;
18     cout<<"res is:"<<c*y;
19
20     return 0;
21 }
22
```

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.

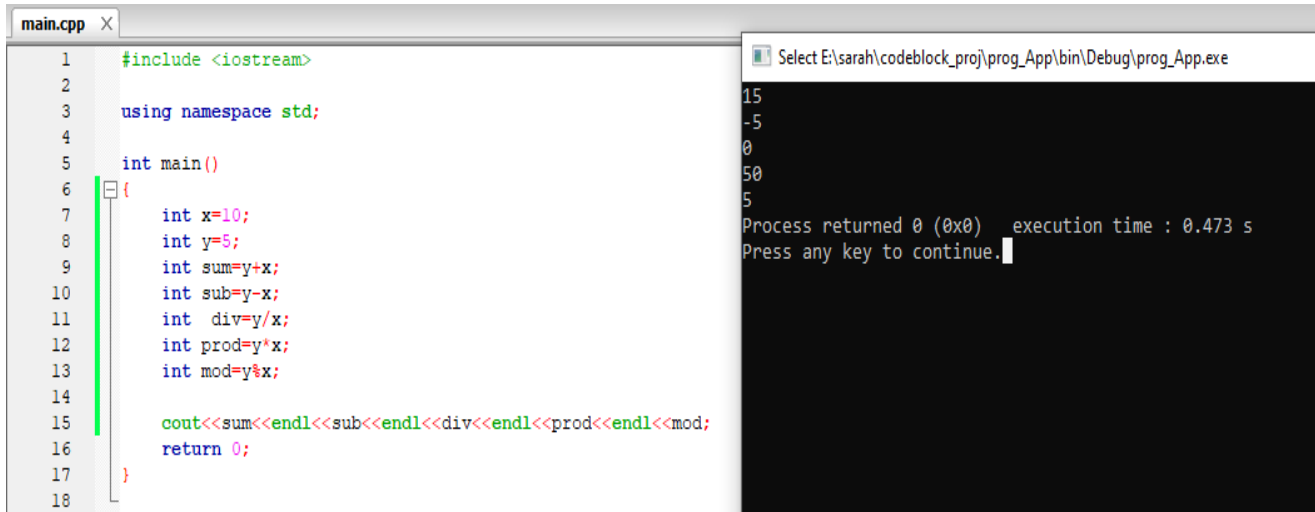
Arithmetic Operators

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)

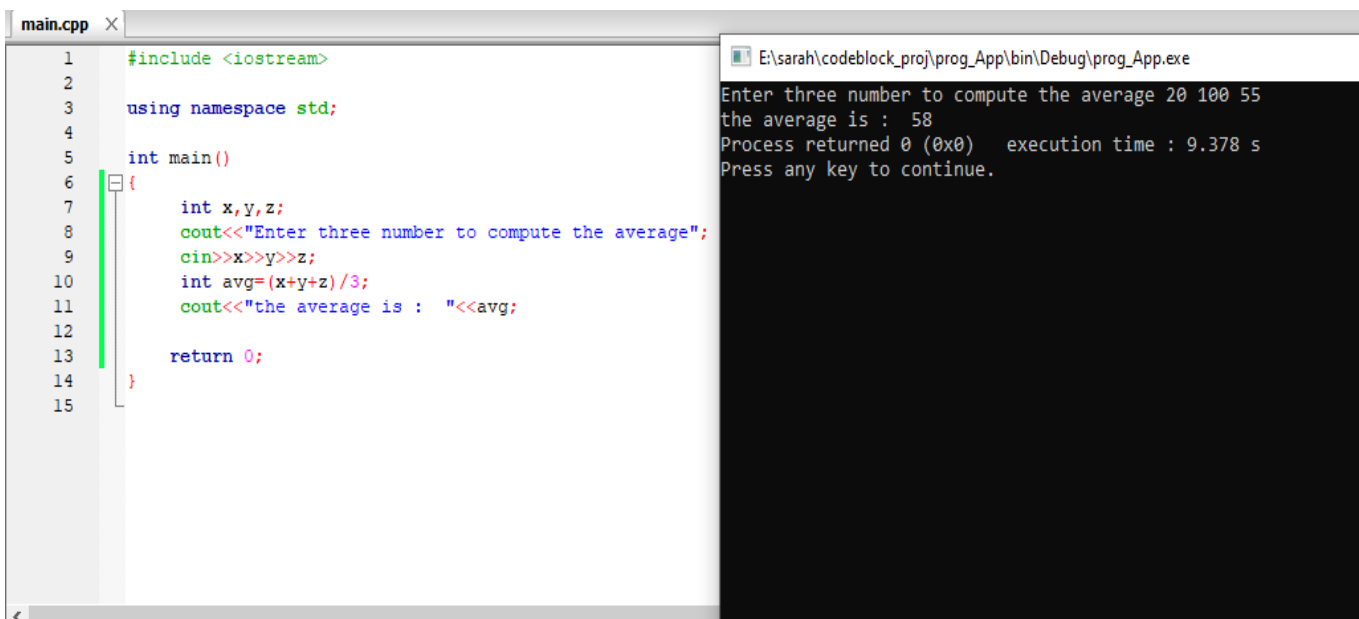
Example



```
main.cpp X
1 #include <iostream>
2
3 using namespace std;
4
5 int main()
6 {
7     int x=10;
8     int y=5;
9     int sum=y+x;
10    int sub=y-x;
11    int div=y/x;
12    int prod=y*x;
13    int mod=y%x;
14
15    cout<<sum<<endl<<sub<<endl<<div<<endl<<prod<<endl<<mod;
16    return 0;
17 }
18
```

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

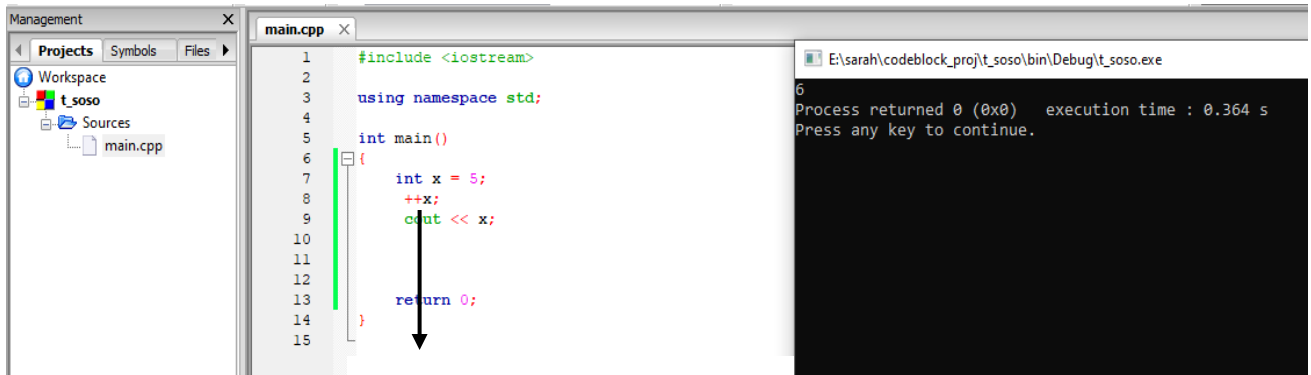


```
main.cpp X
1 #include <iostream>
2
3 using namespace std;
4
5 int main()
6 {
7     int x,y,z;
8     cout<<"Enter three number to compute the average";
9     cin>>x>>y>>z;
10    int avg=(x+y+z)/3;
11    cout<<"the average is : "<<avg;
12
13    return 0;
14 }
15
```

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



```
1 #include <iostream>
2
3 using namespace std;
4
5 int main()
6 {
7     int x = 5;
8     ++x;
9     cout << x;
10
11
12
13     return 0;
14 }
15
```

E:\sarah\codeblock_proj\t_soso\bin\Debug\t_soso.exe
6
Process returned 0 (0x0) execution time : 0.364 s
Press any key to continue.

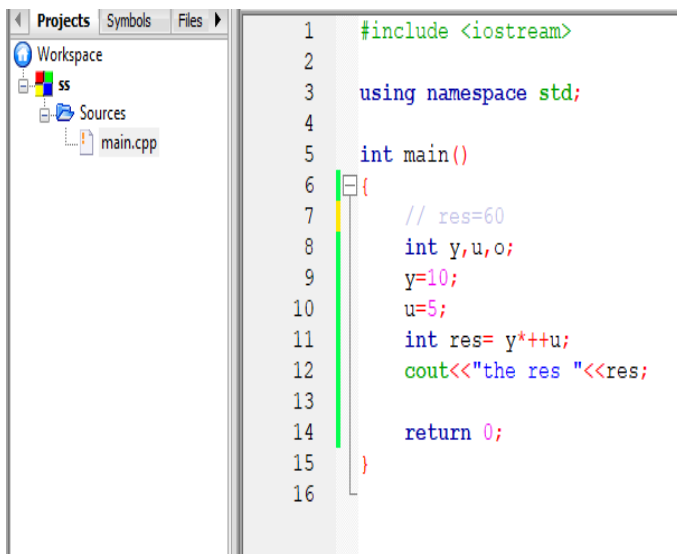
++x or x++ means $x=x+1$

Difference between ++x and x++:

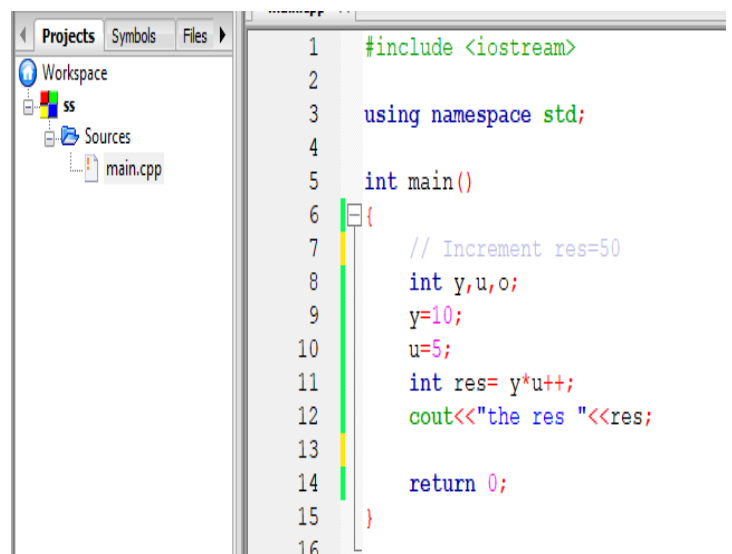
both ++x and x++ are used to increment variable x by 1.

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’.



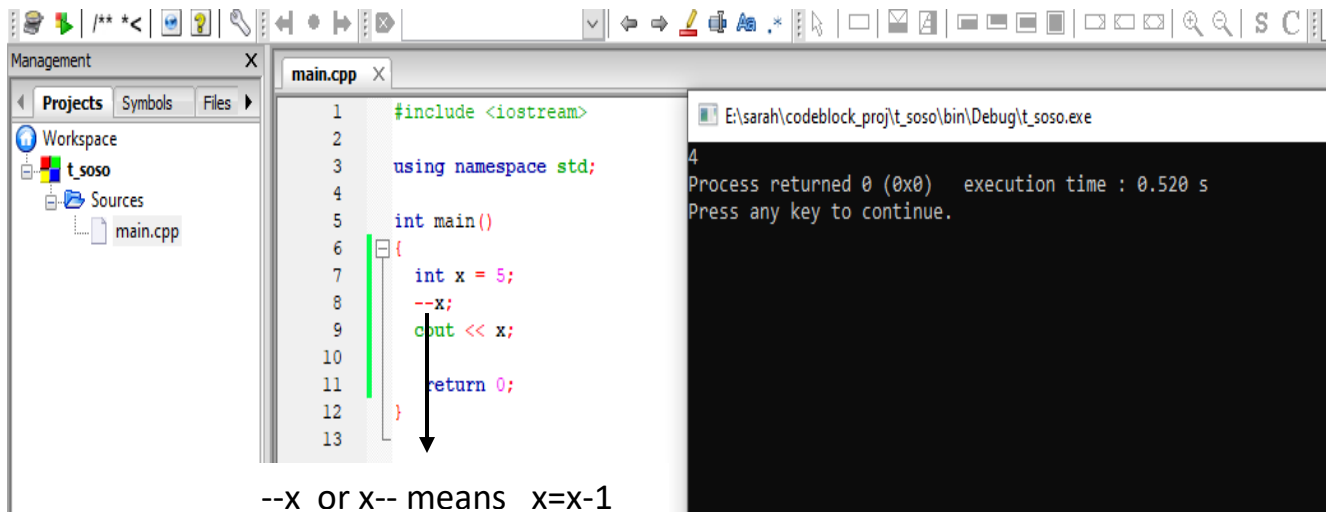
```
1 #include <iostream>
2
3 using namespace std;
4
5 int main()
6 {
7     // res=60
8     int y,u,o;
9     y=10;
10    u=5;
11    int res= y*++u;
12    cout<<"the res "<<res;
13
14    return 0;
15 }
16
```



```
1 #include <iostream>
2
3 using namespace std;
4
5 int main()
6 {
7     // Increment res=50
8     int y,u,o;
9     y=10;
10    u=5;
11    int res= y*u++;
12    cout<<"the res "<<res;
13
14    return 0;
15 }
16
```

Decrement

- ▶ decrease the variable value by 1



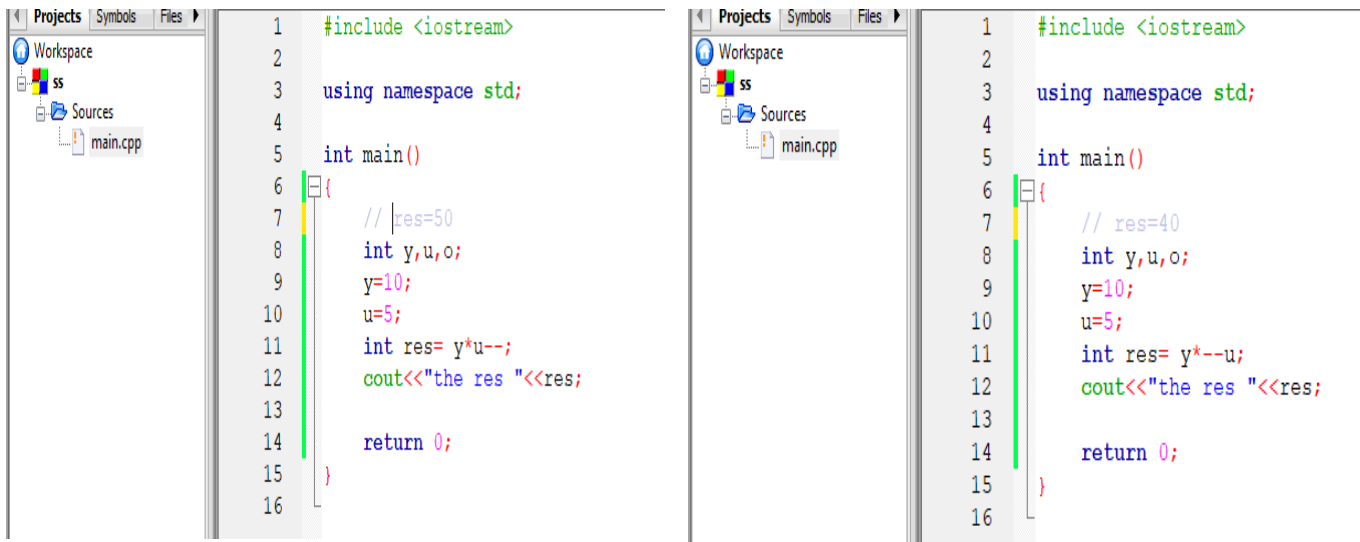
The screenshot shows a code editor with a file named `main.cpp`. The code is as follows:

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main()
6 {
7     int x = 5;
8     --x;
9     cout << x;
10
11     return 0;
12 }
13
```

Below the code, a note states: `--x or x-- means x=x-1`. To the right, a terminal window shows the execution output:

```
E:\sarah\codeblock_proj\t_soso\bin\Debug\t_soso.exe
4
Process returned 0 (0x0)   execution time : 0.520 s
Press any key to continue.
```

Difference between --x and x--



The left screenshot shows a code editor with a file named `main.cpp`. The code is as follows:

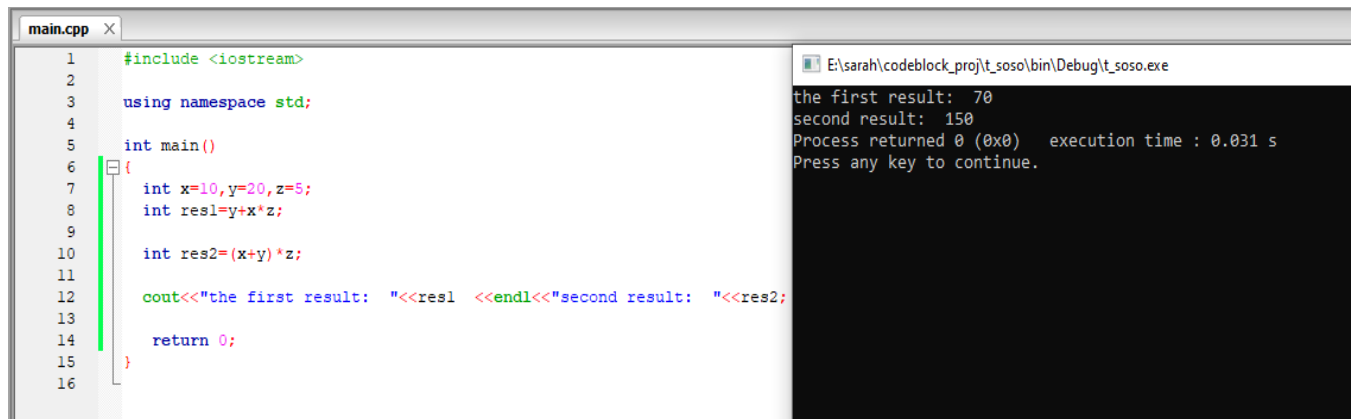
```
1 #include <iostream>
2
3 using namespace std;
4
5 int main()
6 {
7     // res=50
8     int y,u,o;
9     y=10;
10    u=5;
11    int res= y*u--;
12    cout<<"the res "<<res;
13
14    return 0;
15 }
16
```

The right screenshot shows a code editor with a file named `main.cpp`. The code is as follows:

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main()
6 {
7     // res=40
8     int y,u,o;
9     y=10;
10    u=5;
11    int res= y*--u;
12    cout<<"the res "<<res;
13
14    return 0;
15 }
16
```

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

Example:



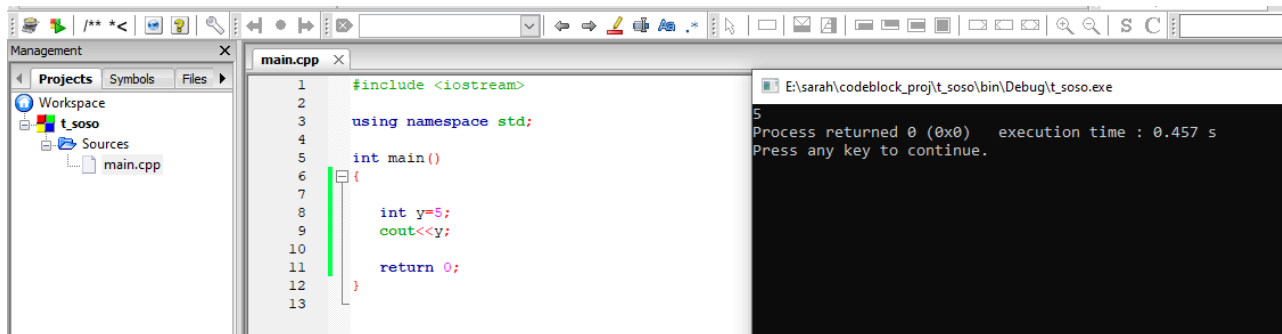
```
main.cpp x
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int x=10,y=20,z=5;
8      int res1=y*x*z;
9
10     int res2=(x+y)*z;
11
12     cout<<"the first result: "<<res1 <<endl<<"second result: "<<res2;
13
14     return 0;
15 }
16
```

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

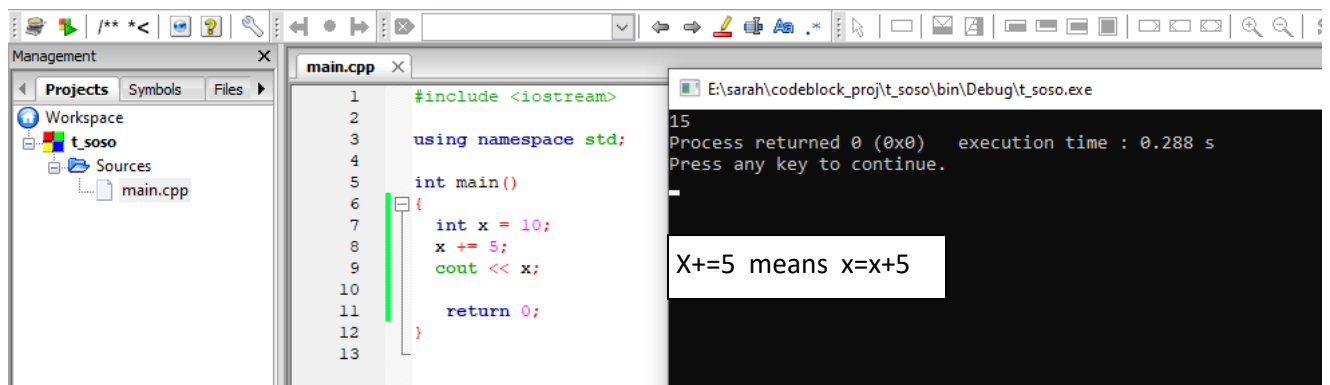
In the example below, we use the **assignment** operator (=) to assign the value **10** to a variable called **x**:



```
main.cpp x
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7
8      int y=5;
9      cout<<y;
10
11     return 0;
12 }
13
```

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



```
main.cpp x
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int x = 10;
8      x += 5;
9      cout << x;
10
11     return 0;
12 }
13
```

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

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

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

A black terminal window with the number 15 displayed in white text.

Example 3:

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

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

A black terminal window with the number 1.66667 displayed in white text.

Example 4:

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

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

A black terminal window with the number 2 displayed in white text.

Comparison Operators

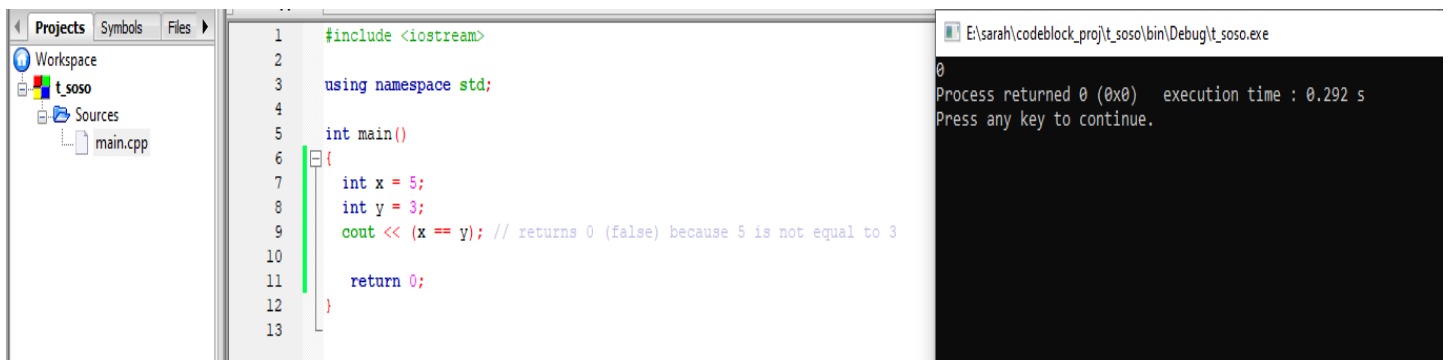
Comparison operators are used to compare two values.

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:



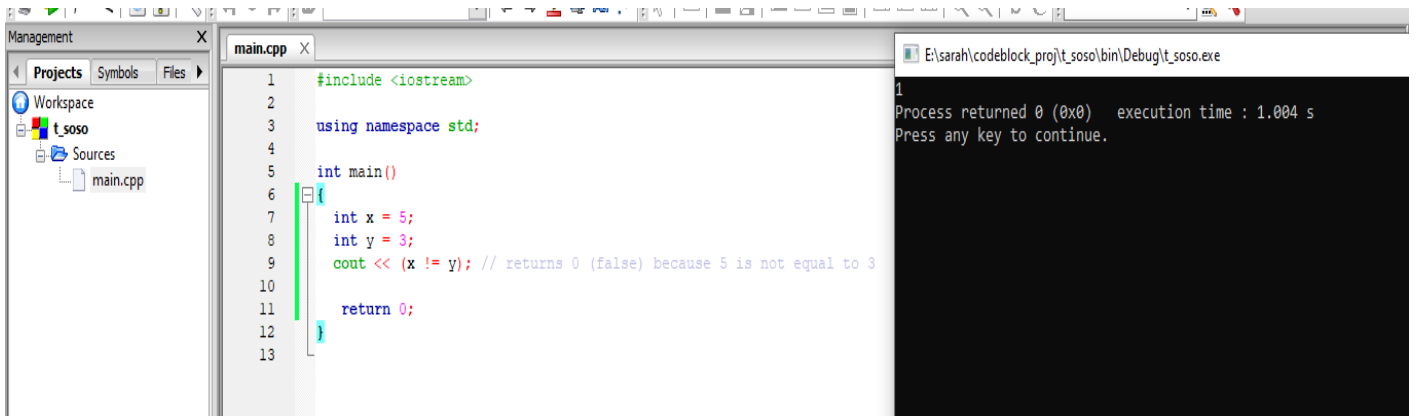
The screenshot shows a code editor with the following C++ code:

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main()
6 {
7     int x = 5;
8     int y = 3;
9     cout << (x == y); // returns 0 (false) because 5 is not equal to 3
10
11     return 0;
12 }
13
```

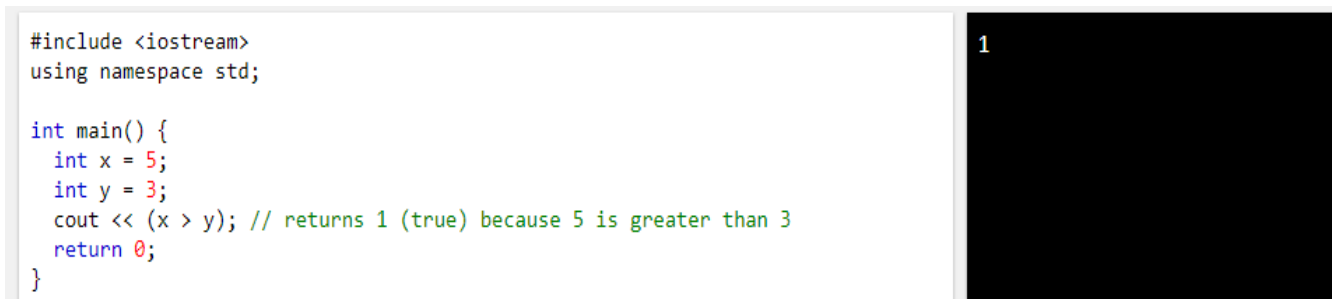
To the right, a terminal window displays the output of the program:

```
E:\sarah\codeblock_proj\t_soso\bin\Debug\t_soso.exe
0
Process returned 0 (0x0)   execution time : 0.292 s
Press any key to continue.
```

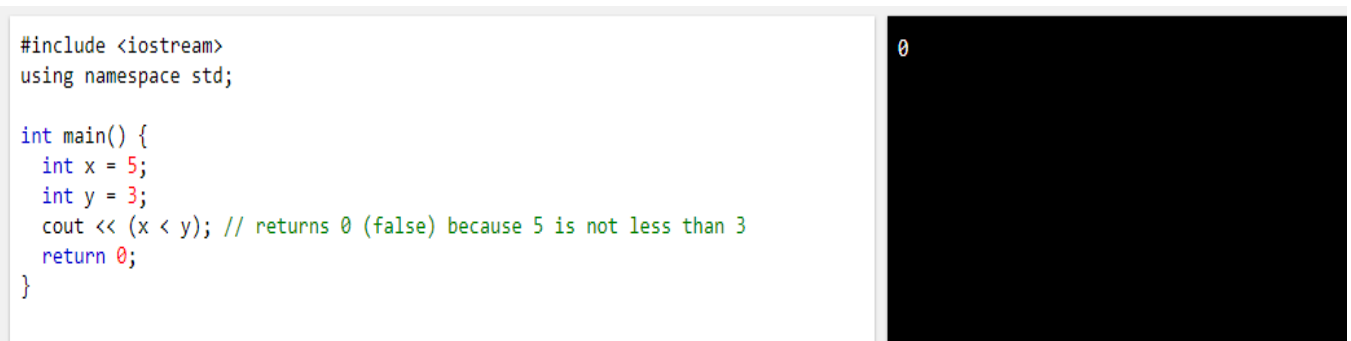
Example2:



Example 3



Example 4:



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 program to check the number negative or positive?

The screenshot shows a code editor with a project named 'sara' and a file named 'main.cpp'. The code in 'main.cpp' is as follows:

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main()
6 {
7     int num;
8     cout<<"enter the number";
9     cin>>num;
10    if(num>0){
11        cout<<"the number is positive";
12    }
13    else{
14        cout<<"the number is negative";
15    }
16 }
17 return 0;
18 }
19 }
```

The execution output in the terminal window is:

```
E:\sarah\codeblock_proj\sara\bin\Debug\sara.exe
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

```
main.cpp x
1 #include <iostream>
2
3 using namespace std;
4
5 int main()
6 {
7     int hours,rate;
8     double gross_pay;
9     cout<<"Enter number of hours and rate";
10    cin>>hours>>rate;
11    if(hours>40){
12        gross_pay = rate * 40 + 1.5 * rate * (hours - 40);
13        cout<<"the gross pay is: " <<gross_pay;
14    }
15    else{
16        gross_pay = rate * hours;
17        cout<<"the gross pay is: " <<gross_pay;
18    }
19
20    return 0;
21
22
23
E:\sarah\codeblock_proj\eee\bin\Debug\eee.exe
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 a program to Calculate students grades?

```
main.cpp x
4
5 int main()
6 {
7     int degree;
8     string grade;
9     cout<<"Enter your degree";
10    cin>>degree;
11    if(degree>=85){
12
13        cout<<"Excellent";
14    }
15    else if(degree>=70){
16
17        cout<<"very good";
18    }
19    else if (degree>=65){
20
21        cout<<"good";
22    }
23    else{
24
25        cout<<"Weak";
26    }
27
28    return 0;
29
30
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 &&

```
main.cpp x | E:\sarah\codeblock_proj\eee\bin\Debug\eee.exe
1 #include <iostream>
2
3 using namespace std;
4
5 int main()
6 {
7     int y=10;
8     int v=-5;
9     if( y>0 &&v>0 ){
10
11         cout<< y/v;
12     }
13     else{
14
15         cout<<" there is a negative value";
16     }
17
18     return 0;
19 }
20
```

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

Example :Or ||

```
main.cpp x | E:\sarah\codeblock_proj\eee\bin\Debug\eee.exe
1 #include <iostream>
2
3 using namespace std;
4
5 int main()
6 {
7     string grade;
8     int score;
9     cout<<"enter grade and score"<<endl;
10    cin>>grade >>score;
11
12    if(grade=="good" || score>=65){
13        cout<<"you passed and go to the next step";
14    }
15    else{
16
17        cout<<"please try again";
18    }
19
20    return 0;
21 }
22
```

```
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

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main()
6 {
7
8     int num1,num2;
9     float res;
10    cout<<"enter num1 and num2"<<endl;
11    cin>>num1>>num2 ;
12
13    if((num1>=10) && !(num2==0)){
14        res=num1/num2;
15        cout<<res;
16    }
17    else{
18
19        cout<<"retype num2 not equal zero";
20    }
21
22    return 0;
23 }
24
```

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

```
main.cpp x
1 #include <iostream>
2
3 using namespace std;
4
5 int main()
6 {
7
8     float num1,num2;
9     float res;
10    cout<<"enter num1 and num2"<<endl;
11    cin>>num1>>num2 ;
12
13    if(num1>=10 && !num2==0){
14        res=num1/num2;
15        cout<<res;
16    }
17    else{
18
19        cout<<"retype num2 not equal zero";
20    }
21
22    return 0;
23 }
24
```

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.
```

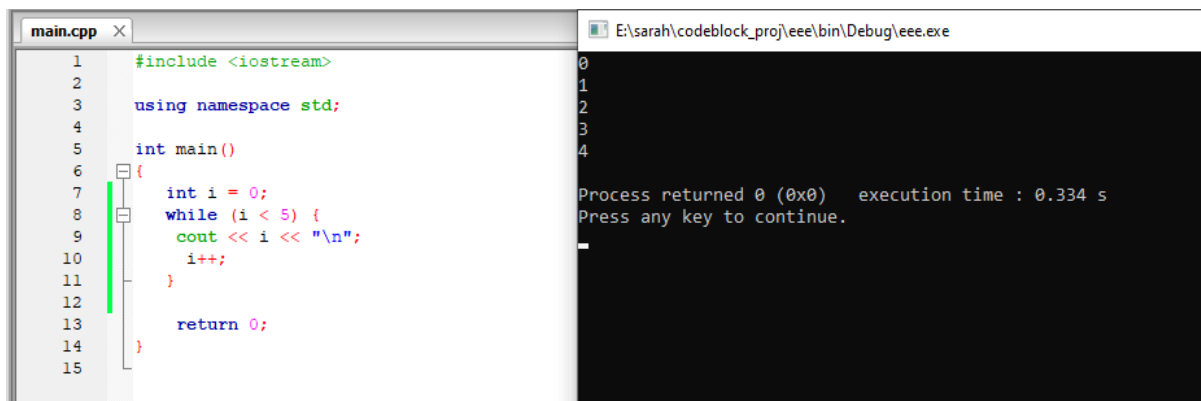


While Loop Operation

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  
}
```

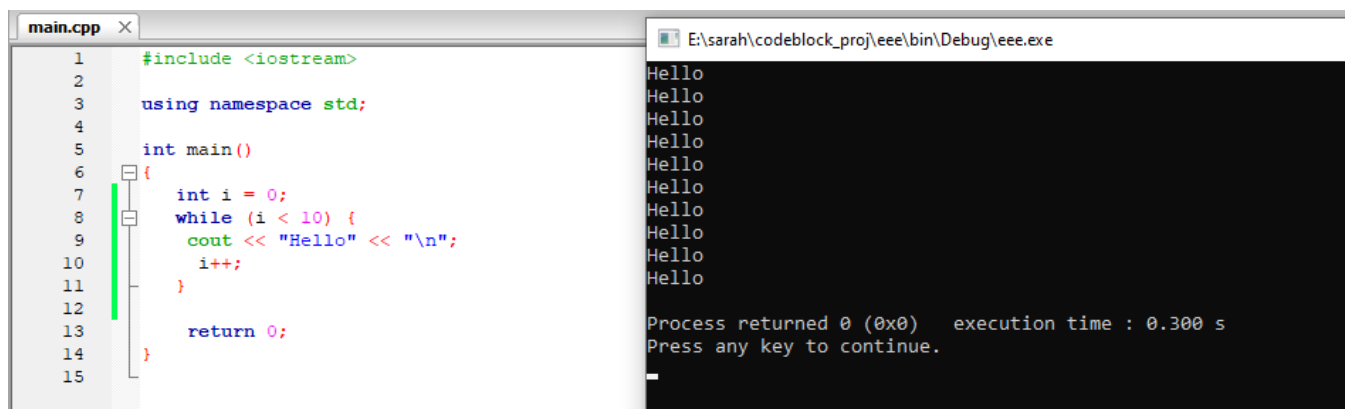


The screenshot shows a code editor window with a file named `main.cpp` and a terminal window. The code in `main.cpp` is as follows:

```
1 #include <iostream>  
2  
3 using namespace std;  
4  
5 int main()  
6 {  
7     int i = 0;  
8     while (i < 5) {  
9         cout << i << "\n";  
10        i++;  
11    }  
12  
13    return 0;  
14 }  
15
```

The terminal window shows the output of the program, which is the numbers 0 through 4, one per line. Below the numbers, the terminal displays the message: "Process returned 0 (0x0) execution time : 0.334 s Press any key to continue."

Write a program print hello 10 times?

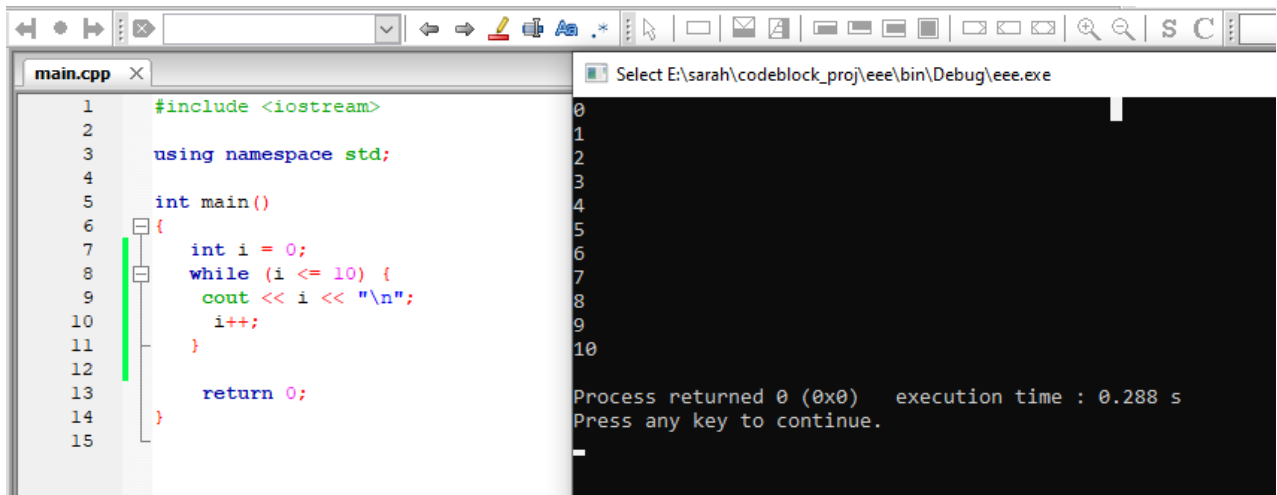


The screenshot shows a code editor window with a file named `main.cpp` and a terminal window. The code in `main.cpp` is as follows:

```
1 #include <iostream>  
2  
3 using namespace std;  
4  
5 int main()  
6 {  
7     int i = 0;  
8     while (i < 10) {  
9         cout << "Hello" << "\n";  
10        i++;  
11    }  
12  
13    return 0;  
14 }  
15
```

The terminal window shows the output of the program, which is the word "Hello" printed 10 times, one per line. Below the "Hello" lines, the terminal displays the message: "Process returned 0 (0x0) execution time : 0.300 s Press any key to continue."

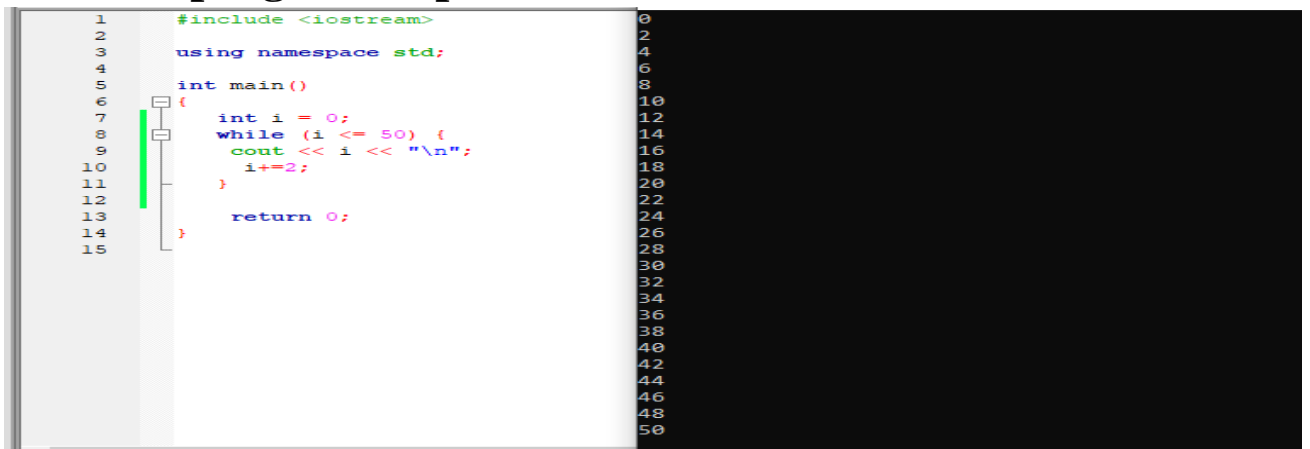
Write a program to print number from 0 to 10?



```
main.cpp x Select E:\sarah\codeblock_proj\eee\bin\Debug\eee.exe
1 #include <iostream>
2
3 using namespace std;
4
5 int main()
6 {
7     int i = 0;
8     while (i <= 10) {
9         cout << i << "\n";
10        i++;
11    }
12
13    return 0;
14 }
15
```

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.

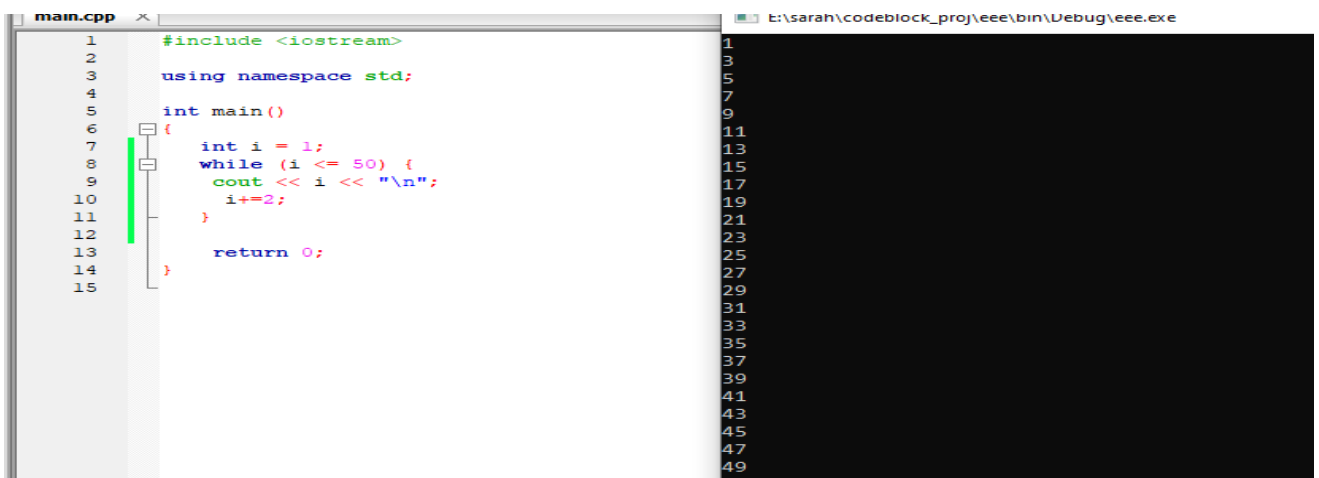
Write a program to print even numbers until 50?



```
1 #include <iostream>
2
3 using namespace std;
4
5 int main()
6 {
7     int i = 0;
8     while (i <= 50) {
9         cout << i << "\n";
10        i+=2;
11    }
12
13    return 0;
14 }
15
```

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 program to print odd numbers until 50?



```
main.cpp x E:\sarah\codeblock_proj\eee\bin\Debug\eee.exe
1 #include <iostream>
2
3 using namespace std;
4
5 int main()
6 {
7     int i = 1;
8     while (i <= 50) {
9         cout << i << "\n";
10        i+=2;
11    }
12
13    return 0;
14 }
15
```

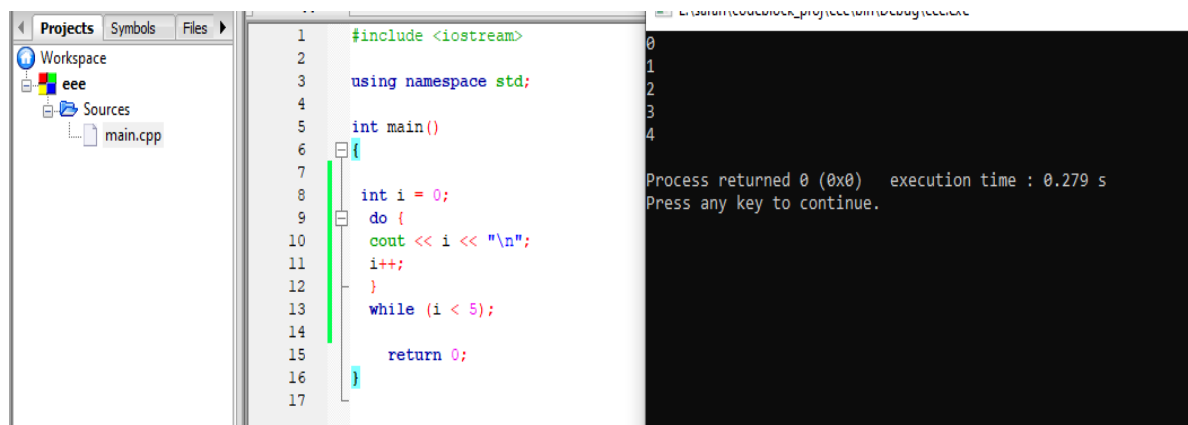
1
3
5
7
9
11
13
15
17
19
21
23
25
27
29
31
33
35
37
39
41
43
45
47
49

do-while loop

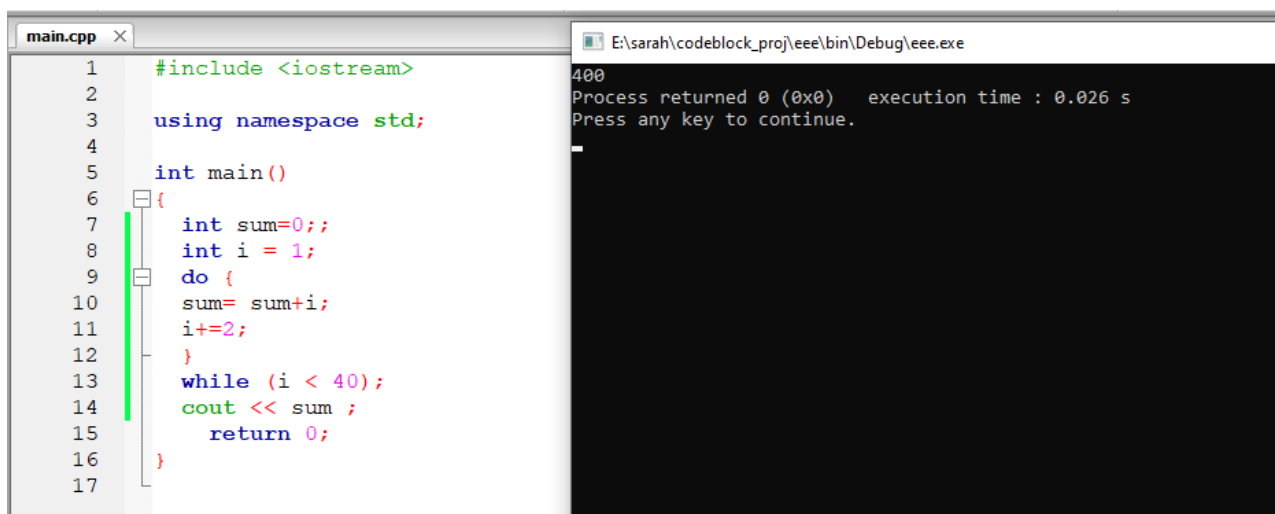
The **do/while** loop is a variant of the **while** loop. This loop will execute the code block once, before checking if the condition is true, then it will repeat the loop as long as the condition is true.

```
Syntax

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



Write a program to print sum of odd numbers <40?



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 a program 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?

