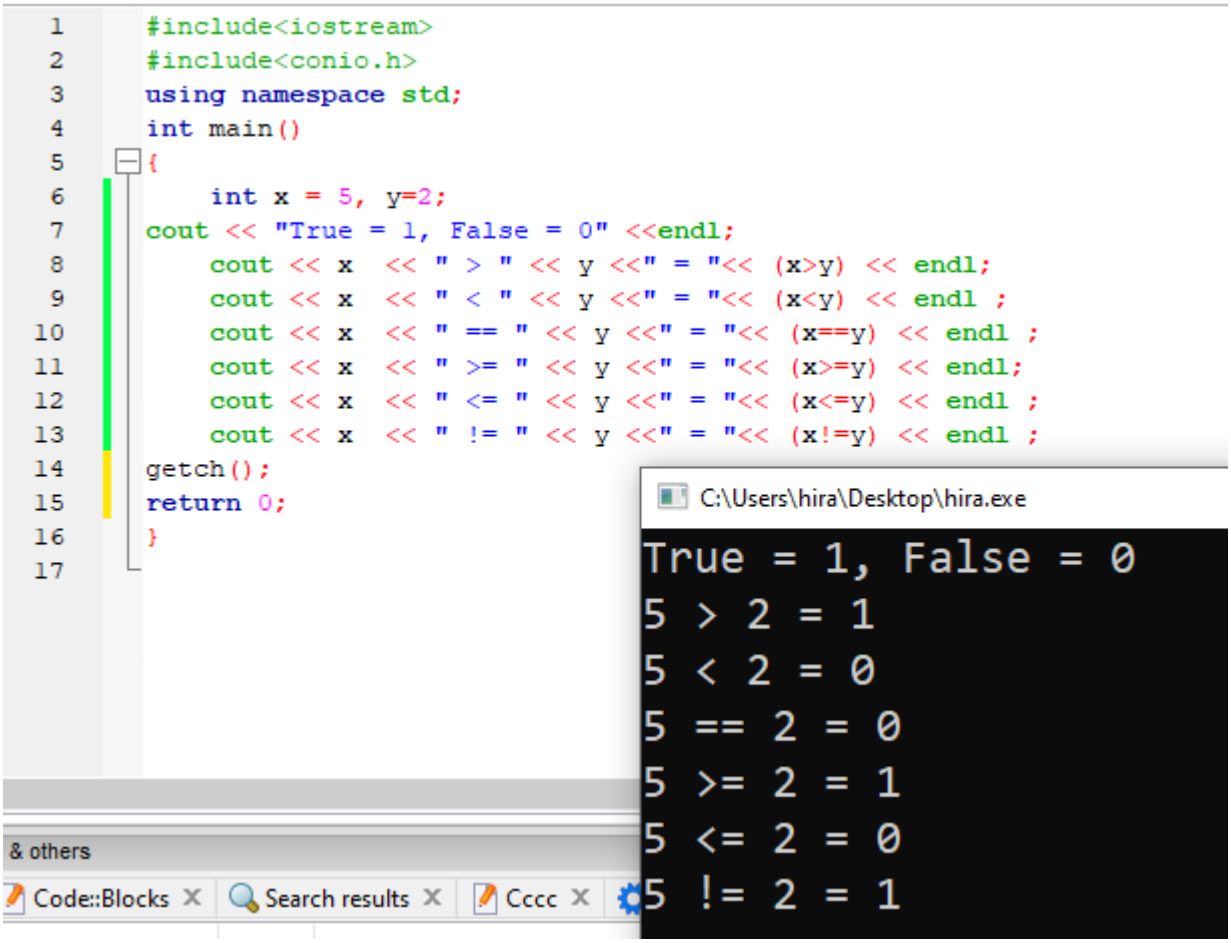
2. Relational /Comparison Operators (6)

- Relational operators are used for checking conditions whether the given condition is true or false. If the condition is true, it will return non-zero value, if the condition is false, it will return 0.
- Example :- Suppose we have, int X = 5, Y = 2; then –

Operator	Name	Description	Example
>	Greater than	Check whether the left operand is greater than right operand or not.	(X > Y) will return true
<	Smaller than	Check whether the left operand is smaller than right operand or not.	(X < Y) will return false
>=	Greater than or Equal to	Check whether the left operand is greater or equal to right operand or not.	(X >= Y) will return true
<=	Smaller than or Equal to	Check whether the left operand is smaller or equal to right operand or not.	(X <= Y) will return false
==	Equal to	Check whether the both operands are equal or not.	(X == Y) will return false
!=	Not Equal to	Check whether the both operands are equal or not.	(X != Y) will return true



```cpp
#include<iostream>
#include<conio.h>
using namespace std;
int main()
{
    int x = 5, y=2;
cout << "True = 1, False = 0" <<endl;
    cout << x  << " > " << y <<" = "<< (x>y) << endl;
    cout << x  << " < " << y <<" = "<< (x<y) << endl ;
    cout << x  << " == " << y <<" = "<< (x==y) << endl ;
    cout << x  << " >= " << y <<" = "<< (x>=y) << endl;
    cout << x  << " <= " << y <<" = "<< (x<=y) << endl ;
    cout << x  << " != " << y <<" = "<< (x!=y) << endl ;
getch();
return 0;
}
```

```
C:\Users\hira\Desktop\hira.exe
True = 1, False = 0
5 > 2 = 1
5 < 2 = 0
5 == 2 = 0
5 >= 2 = 1
5 <= 2 = 0
5 != 2 = 1
```

& others

Code::Blocks × Search results × Cccc ×

3. Increment/Decrement Operators (2)

- C++ allows two very useful operators not generally found in other computer languages.
- These are the Increment (++) and Decrement (– –) operators. The operation ++ adds 1 to its operand, and – – subtracts 1.
- It is also called unary operator.
- Example:- Assume a=10; then-
  - ✓ a++ (post increment) = (10+1) = 11
  - ✓ ++a (pre increment) = (11+1) = 12
  - ✓ a-- (post decrement) = (12-1) = 11
  - ✓ --a (pre decrement) = (11-1) = 10

Solve

1. int a =7 , b =5, c= 10;
   bool result = (a+b*c > 50);
2. int x = 5;
   int y = ++x + x++ + --x;
3. int a =8, b=3;
   bool r = ((a/b) * b ==a);

4. int a=3;
   int b= a++ + ++a + a--;
5. int a = 6, b = 9, c =2;
   bool result = (a*b/c <= b+c);
6. int d =10;
   int e= d-- - --d + ++d + d++;
7. int a=2, b=3;
   bool r = (a++ *b-- > --a + ++b);
8. int a =4, b=5;
   bool r = (a++ + ++b > 10);
9. int a=3;
   int b= a++ + ++a + a--;
   bool r= (b == 11);
10. int x=5, y=6;
    bool new=(x++ *2 < --y +7);

Answer

1→ 1, 2→18, 3→0, 4→ 13, 5→0, 6→20, 7→1, 8→0, 9→0, 10→Error,(error solve answer→ 1)

```
#include<iostream>
#include<conio.h>
using namespace std;
int main()
{
    int a=5;
    cout << "a = " <<a << endl;
    cout << "a++ then value of a = " << a++ << endl;
    cout << "++a then value of a = " << ++a << endl;
    int b=10;
    cout << "b = " <<b <<endl;
    cout << "b-- then value of b = "  << b-- << endl;
    cout << "--b then value of b = "  << --b << endl;
    getch();
return 0;
}
```

```
C:\Users\hira\Desktop\hira.exe
a = 5
a++ then value of a = 5
++a then value of a = 7
b = 10
b-- then value of b = 10
--b then value of b = 8
```

& others

Code::Blocks | Search results | Cccc | Bu

4. Logical Operators (5)

- Logical operators perform logical operations on Boolean expressions (expressions that evaluate to either true or false).
- There are three logical operators in C++:

Suppose we have,

int X = 5, Y = 2;

Operator	Name	Description	Example
&&	AND	Return true if all conditions are true, return false if any of the condition is false.	if(X > Y && Y < X) will return true
||	OR	Return false if all conditions are false, return true if any of the condition is true.	if(X > Y || X < Y) will return true
!	NOT	Return true if condition if false, return false if condition is true.	if(!(X>y)) will return false

True & False

AND (&&) Table:

A	B	A && B
T	T	T
T	F	F
F	T	F
F	F	F

OR (||) Table:

A	B	A || B
T	T	T
T	F	T
F	T	T
F	F	F

NOT (!) Table:

A	!A
T	F
F	T

1 & 0

AND (&&) Table:

A	B	A && B
1	1	1
1	0	0
0	1	0
0	0	0

OR (||) Table:

A	B	A || B
1	1	1
1	0	1
0	1	1
0	0	0

NOT (!) Table:

A	!A
1	0
0	1

Operator	Name	Example
&&	Logical And	5 < 6 && 6 < 6 // gives 0
||	Logical Or	5 < 6 || 6 < 5 // gives 1
!	Logical Negation (Not)	!(5 == 5) // gives 0

```cpp
#include<iostream>
#include<conio.h>
using namespace std;
int main()
{
    int a= 5, b= 10, c=15;
    bool r = (a>b) && (b>c);
    bool s = (a<b) && (b<c);
    bool t  = (a>b) ||  (b<c);

    cout << r << endl ;
    cout <<  s << endl;
      cout <<  t << endl;


    getch();
return 0;
}
```

```
C:\Users\hira\Desktop\hi
0
1
1
```

Solve

1. int a = 10, b=5, c=20;
   cout << (a>b && c>a);
2. int x =4;
   cout << (x++ > 4 || ++x ==6);
3. int a =0, b=5;
   cout << !(a+b);
4. int x=1, y=2;
   cout << (++x && y++);
5. int a=10, b=3;
   cout << (a %b ==1);
6. int x=0, y=5;
   cout << (x++ || y++);
7. int a=5, b=10, c=15;
   cout << (a+b>c && b++ <c);
8. int x=5;
   cout << (--x && x--);
9. int main()
   {
   int a = 21;
   int c ;
   c = a++;
   cout << c;
   return 0;
   }

Answer

1→1, 2→1, 3→0, 4→1, 5→1, 6→ 1, 7→ 0, 8 → 1, 9 → 21

==========HIRA KUMAR==========