

Introduction to Programming

Lecture Conditional Statements

Muhammad Salman

ABASYN
UNIVERSITY

Summary

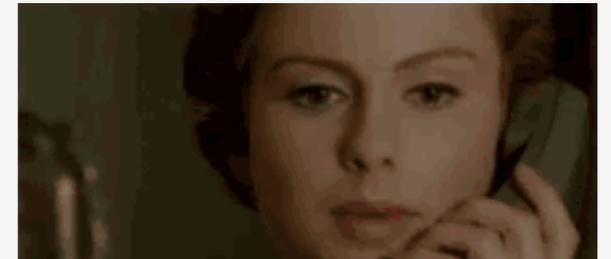
- Conditional statements
- Flow Chart
 - First Program
- if/else structure
- Nested if
- Logical Operator
 - Second Program

We often make decisions based on some condition

→ Mom, bring some drinking water while coming back if the shop is open.



Okay, honey, I'll bring it if the shop is open.



There is a condition for decision-making

She will bring water if the shop is open. And if the shop is closed, she will come back home without water.

Conditional Statements

- The statements of a computer program are executed one after another in the order in which they are written.
- The order of execution of the statements in a program can be changed.
- This is done with the help of conditional statements.(else block)
- The conditional statements are used to execute or ignore a set of statements after testing a condition. The conditional statements are also called selection statements.

Relational Experations

- A relational expression consists of constants, variables, or arithmetic expressions that are combined by a **relational operator**.
- A relational expression is written to find a relation between to expressions. **It returns only one value which is either true or false.**
 - for example, **10>9** is a relational expression it indicates a relation between 10 and 9.
- **Since the 10 is greater then 9 therefor it returns the true value.**

Types of operators

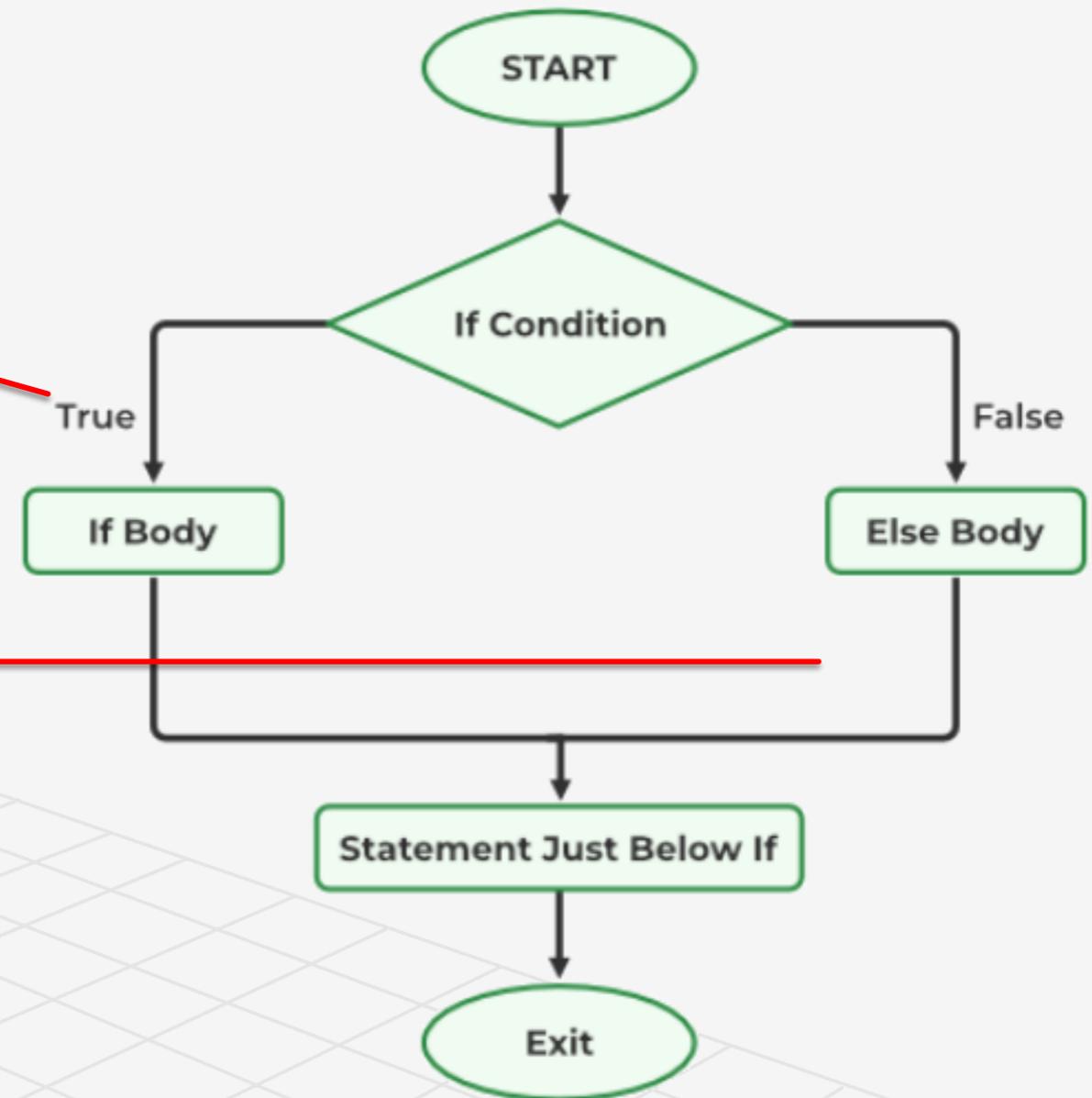
Relational operator

Relational operators facilitate condition testing, allowing us to create and assign variable values. For example, if A equals 45 and B equals 50, you might write $A < B$ or A is less than B. That $<$ symbol is a relational operator that produces true or false results. Familiar relational operators include:

- Assignment `=` `int A = 45, B = 50`
- Equivalence `==`
- Less than `<`
- Greater than `>`
- Less than or equal to `<=`
- Greater than or equal to `>=`
- Does not equal `!=`

How it works .. ???

```
If (condition)
{
  Statement_1 ;
  Statement_1;
  Statement_1;
}
else
{
  Statement_1 ;
  Statement_1;
  Statement_1;
}
```



How it works .. ???

a>b

a<b

a==b

Let's test if these three conditions

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

int main() {
    int a =100;
    int b =150;
    if (a>b)
        cout <<a <<" is greater then "<<b;
    else
        cout <<b <<" is Greater than "<<a;

    return 0;
}
```

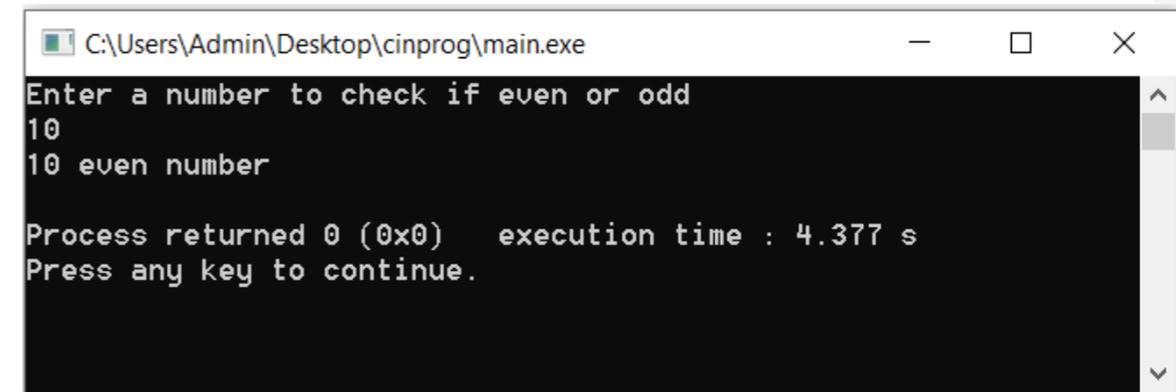
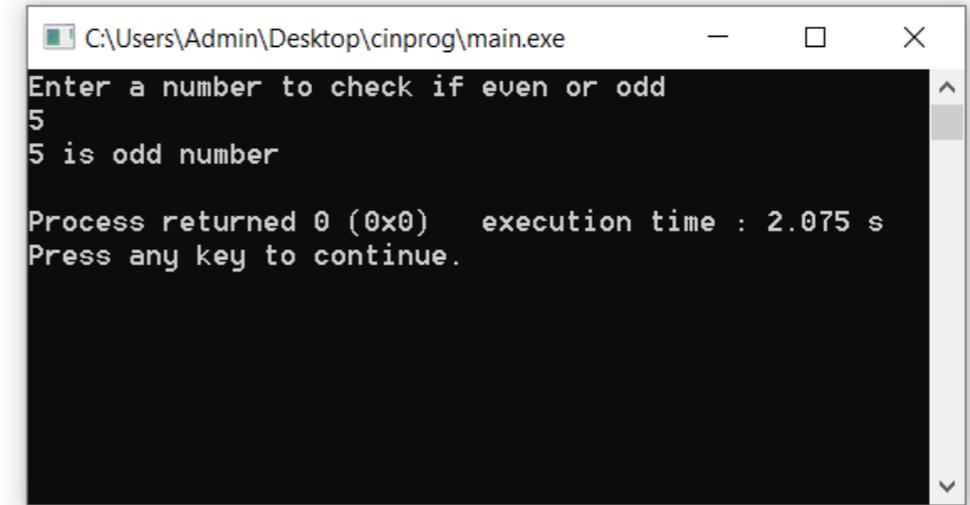
Output

150 is Greater than 100 ???

=== Code Execution Successful ===

Write a Programm to find Odd and Even number

```
main.cpp x
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int i =0;
6      cout<< "Enter a number to check if even or odd"<<endl;
7      cin >>i;
8      if (i == 0)
9      {
10         cout << i << " None" << endl;
11     }
12     else if (i%2==0)
13     {
14         cout <<i <<" even number"<<endl;
15     }
16     else if (i%2 !=0)
17     {
18         cout << i << " is odd number"<<endl;
19     }
20 else {
21     cout << "This statement is always executed.";
22 }
23 return 0;
24 }
25
```



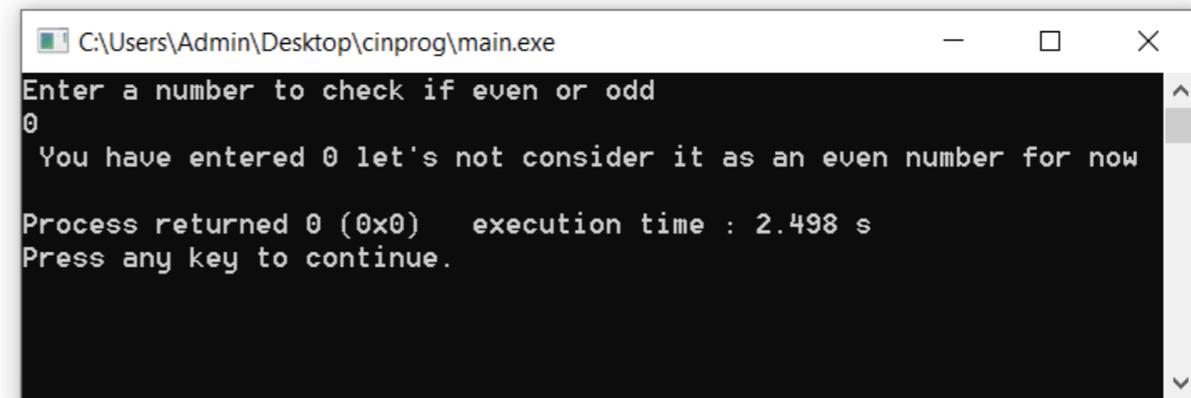
Note !!!. “0 (Zero) is an even number” but for the sake of understanding of the program I don’t want Zero. Ref next slide

Zero is an even Number

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

int main() {
    int i;
    cout << "Enter a number to check if even or odd" << endl;
    cin >> i;
    if (i == 0)
    {
        cout << " You have entered " << i << " let's not consider"
            << " it as an even number for now" << endl;
    }
    else if (i%2 ==0)
    {
        cout << i << " is Even number" << endl;
    }
    else if (i%2 !=0)
    {
        cout << i << " is Even number" << endl;
    }

    else {
        cout << "This statement is always executed.";
    }
    return 0;
}
```



```
C:\Users\Admin\Desktop\cinprog\main.exe
Enter a number to check if even or odd
0
You have entered 0 let's not consider it as an even number for now
Process returned 0 (0x0)   execution time : 2.498 s
Press any key to continue.
```

Positive Number

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

int main() {
    int x ;
    int y ;
    cout <<"Enter x"<<endl;
    cin>> x;
    cout<<"Enter y"<<endl;
    cin >>y;

    if (x > 0)
    {

        cout <<x<< " X is a Positive number"<<endl;
    }

    else if (y > 0)
    {
        cout << y<< " Y is positive " << endl;
    }

    return 0;
}
```

```
C:\Users\Admin\Desktop\cinprog\main.exe
Enter x
5
Enter y
8
5 X is a Positive number

Process returned 0 (0x0)   execution time : 3.933 s
Press any key to continue.
```

```
C:\Users\Admin\Desktop\cinprog\main.exe
Enter x
0
Enter y
10
10 Y is positive

Process returned 0 (0x0)   execution time : 7.236 s
Press any key to continue.
```

Username and Password authentication

main.cpp	Output
<pre>1 // Online C++ compiler to run C++ program online 2 #include <iostream> 3 using namespace std; 4 5 int main() { 6 string name="Abasyn"; 7 string passwd="s2025@n"; 8 string u_name ; 9 string u_passwd; 10 cout <<"Enter your user name here :"<<endl; 11 cin>>u_name; 12 cout <<"Enter your Password here : "<<endl; 13 cin >>u_passwd; 14 if (u_name == name){ 15 if (u_passwd == passwd) 16 cout<<" Welcom you are logged !!! "; 17 } 18 else 19 cout <<"Invalid username or Password"; 20 21 return 0; 22 }</pre>	<pre>Enter your user name here : Abasyn Enter your Password here : s2025@n Welcom you are logged !!! === Code Execution Successful ===</pre>

Nested if Username and Password authentication

main.cpp



Share

Run

Output

```
1 // Online C++ compiler to run C++ program online
2 #include <iostream>
3 using namespace std;
4
5 int main() {
6     string name="Abasyn";
7     string passwrd="s2025@n";
8     string u_name ;
9     string u_passwrd;
10    cout <<"Enter your user name here :"<<endl;
11    cin>>u_name;
12    cout <<"Enter your Password here : "<<endl;
13    cin >>u_passwrd;
14    if (u_name == name){
15        if (u_passwrd == passwrd)
16            cout<<" Welcom you are logged !!! ";
17    }
18    else
19        cout <<"Invalid username or Password";
20
21    return 0;
22 }
```

Enter your user name here :

abasyn

Enter your Password here :

S2025@n

Invalid username or Password

=== Code Execution Successful ===

Nested if

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

int main() {
    int num;
    cout << "Enter an integer: ";
    cin >> num;

    if (num != 0) { // Outer if with braces
        if (num > 0)
            cout << "The number is positive." << endl;
        else
            cout << "The number is negative." << endl;
    } else {
        cout << "The number is 0 and it is neither positive nor negative
        ." << endl;
    }

    return 0;
}
```

Output

```
Enter an integer: 99
The number is positive.
```

Output

```
Enter an integer: -4
The number is negative.
```

Output

```
Enter an integer: 0
The number is 0 and it is neither positive nor negative.
```

Increment & Decrement Operator

Increment:

The Increment operator used to add 1 to the value of a variable is called the increment operator.

The increment operator is represented by a double plus (++) sign. It is used to add 1 to the value of an integer variable. This operator can be used before the variable name.

$xy = xy+1$

```
1 // Online C++ compiler to run C++ program online
2 #include <iostream>
3 using namespace std;
4
5 int main() {
6     int a = 5;
7     int b=5;
8     int x=5;
9     int sum = a+b + (x++);
10
11     cout <<"The sum of a+b+x = ;" <<sum<<endl;
12     cout <<"The valus of x is now = " <<x;
13     return 0;
14 }
```

Output

```
The sum of a+b+x = ;15
The valus of x is now = 6
```

Increment & Decrement Operator

Decrement (--):

The Decrement operator is represented by a double minus (--) sign. It is used to subtract 1 from the values of an integer variable.

For example, to subtract 1 from the values of a variable `xy` the decrement statement is written as `xy--` or `x--`

$$xy = xy - 1$$

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

int main() {
    int a = 5;
    int b=5;
    int x=5;
    int sum = a+b + (--x);

    cout <<"The sum of a+b+x = " <<sum<<endl;
    cout <<"The valus of x is now = " <<x;
    return 0;
}
```

Output

```
The sum of a+b+x = 14
The valus of x is now = 4
```

Increment & Decrement Operator

Decrement (--):

The Decrement operator is represented by a double minus (--) sign. It is used to subtract 1 from the values of an integer variable.

For example, to subtract 1 from the values of a variable `xy` the decrement statement is written as `xy--` or `x--`

$$xy = xy - 1$$

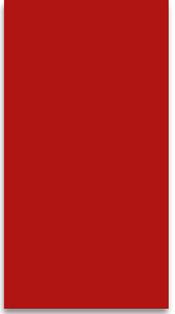
```
#include <iostream>
using namespace std;

int main() {
    int a = 5;
    int b=5;
    int x=5;
    int sum = a+b + (--x);

    cout <<"The sum of a+b+x = " <<sum<<endl;
    cout <<"The valus of x is now = " <<x;
    return 0;
}
```

Output

```
The sum of a+b+x = 14
The valus of x is now = 4
```



Thanks