

Programming (1)

C++

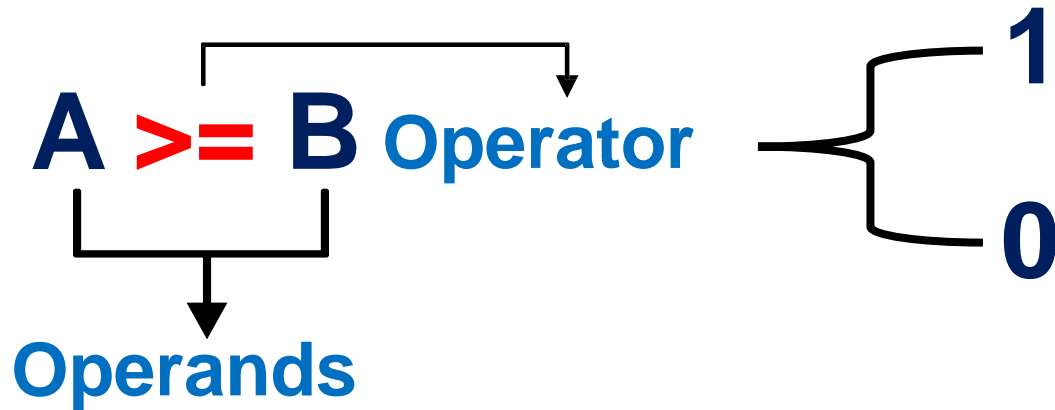
Faculty of Computers and Information
Fayoum University
2020-2021



Categories of Operators in C++

- **Assignment Operator (=)**
- **Arithmetic Operators (+ , - , / , * , %)**
- **Arithmetic Assignment Operators (+= , -= , /= , *= , %=)**
- **Increment and Decrement Operators (++ , --)**
- **Relational Operators (> , < , >= , <= , == , !=)**
- **Logical Operators (&& , || , !)**
- **Conditional Operator (?:)**
- **Bitwise Operators (& , | , ~ , ^ , >> , <<)**

Relational Operators [Comparison]



- For Example:

5 **>=** **6**

5 **!=** **6**

8 **==** **8**

5 **<** **6**

Relational Operators in C++

- They compare two operands and return 0 or 1. If comparison is successful, they return 1 else return 0. They are used to create conditions. One relational operator creates one condition.

Operation	Description	Type	Example
>	Greater Than	Binary	a > b
<	Less Than	Binary	a < b
>=	Greater Than or Equals To	Binary	a >= b
<=	Less Than or Equals To	Binary	a <= b
==	Equals To	Binary	a == b
!=	Not Equals To	Binary	a != b

Relational Operators in C++

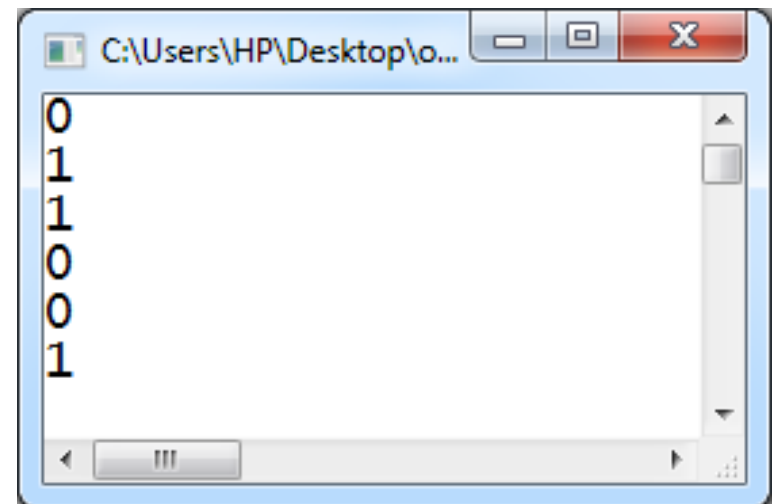
```
#include<iostream>
#include<conio.h>

using namespace std;

int main()
{
    int a=5, b=8, c=-8, d=61, e=15;

    cout<< (a > b) <<endl;
    cout<< (c < a) <<endl;
    cout<< (d >= e) <<endl;
    cout<< (e <= a) <<endl;
    cout<< (c == b) <<endl;
    cout<< (d != e);

    getch();
    return 0;
}
```



Logical Operators in C++

- They perform logical operations on the logical operands. They return either 0 or 1.
- They combine multiple conditions to form one composite condition.

Operation	Description	Type	Example
&&	Logical AND	Binary	a && b
 	Logical OR	Binary	a b
!	Logical NOT	Unary	! a

Logical Operators in C++

a	b	a && b
0	0	0
0	1	0
1	0	0
1	1	1

a	b	a b
0	0	0
0	1	1
1	0	1
1	1	1

a	! a
0	1
1	0

Logical Operators in C++

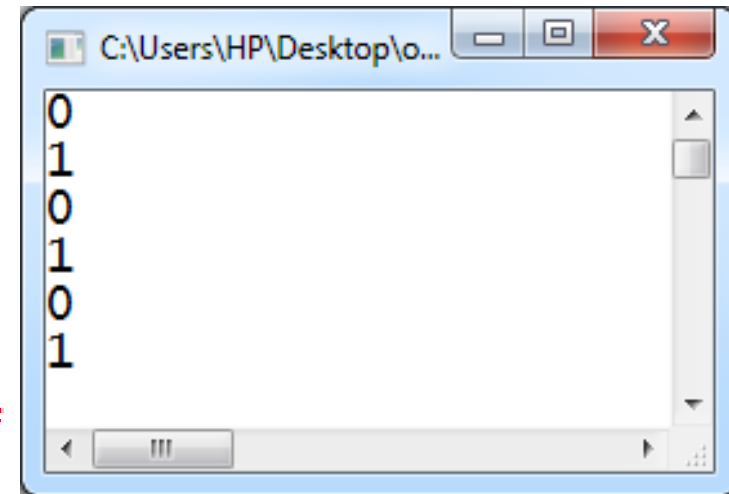
```
#include<iostream>
#include<conio.h>

using namespace std;

int main()
{
    int a=5, b=8, c=-8, d=61, e=15;

    cout<< (a > b && c != d) <<endl;
    cout<< (b < c || d >= e) <<endl;
    cout<< (a > b) <<endl;
    cout<< !(a > b) <<endl;
    cout<< (a == b) <<endl;
    cout<< !(a == b) <<endl;

    getch();
    return 0;
}
```




Logical Operators in C++

- Logical AND is also called as **Short Circuit AND** operator.
 - If the first operand is false (0) then it will not check the second operand.
- Logical OR is also called as **Short Circuit OR** operator.
 - If the first operand is true (1) then it will not check the second operand.


```
int num1 = 5, num2 = 10, num3 = 6, num4 = 15;
```

```
cout << num1 > num2 && num3 < num4;
```



This will not be checked

```
cout << num1 < num2 || num3 != num4;
```



This will not be checked



Program Examples

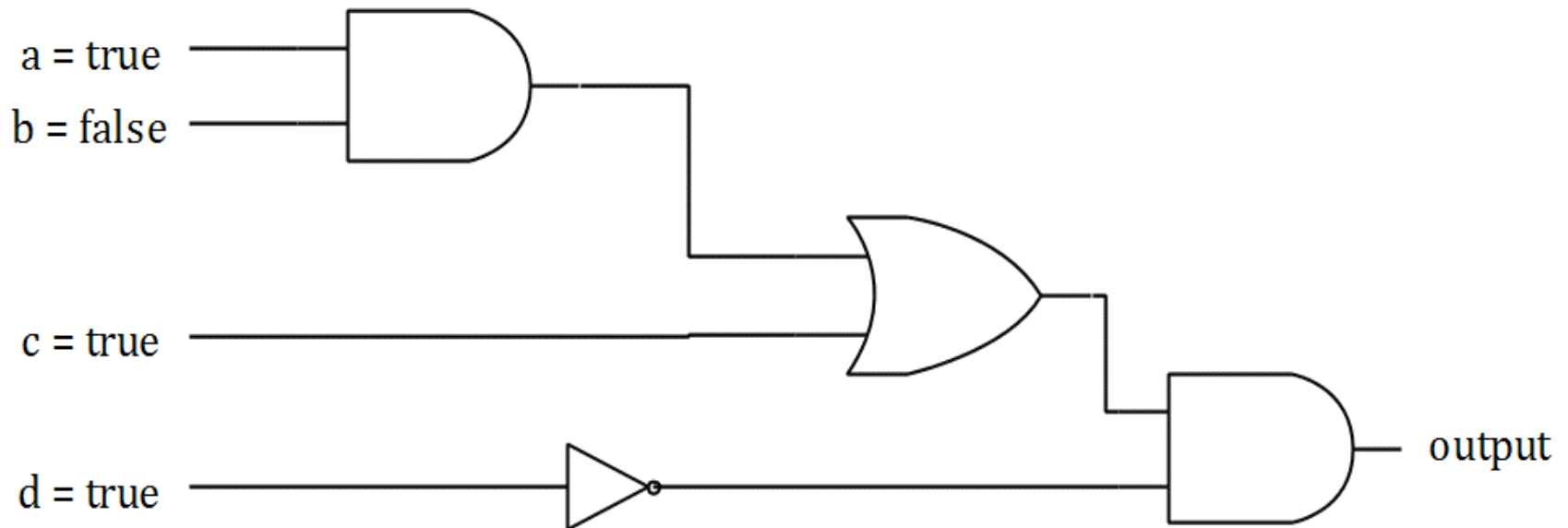
Logic circuit diagrams and logical operators in C++

Program Example 01

Problem Statement:

Write a logic circuit solver program in C++ that solves the following logic circuit.

Where all the inputs are logic switches (which can be either Turned ON (True, 1) or Turned OFF (False, 0)). The program should display the output value



Program Example 01

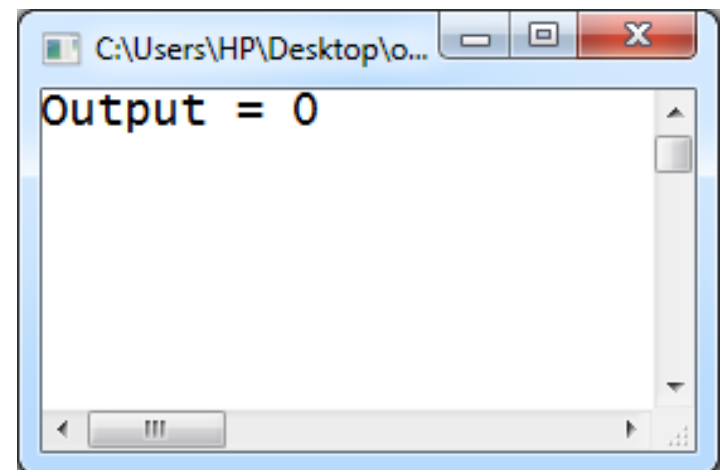
```
#include<iostream>
#include<conio.h>

using namespace std;

int main()
{
    bool a=true, b=false, c=true, d=true, output;

    output = ((a && b) || c) && !d;
    cout<<"Output = "<<output;

    getch();
    return 0;
}
```

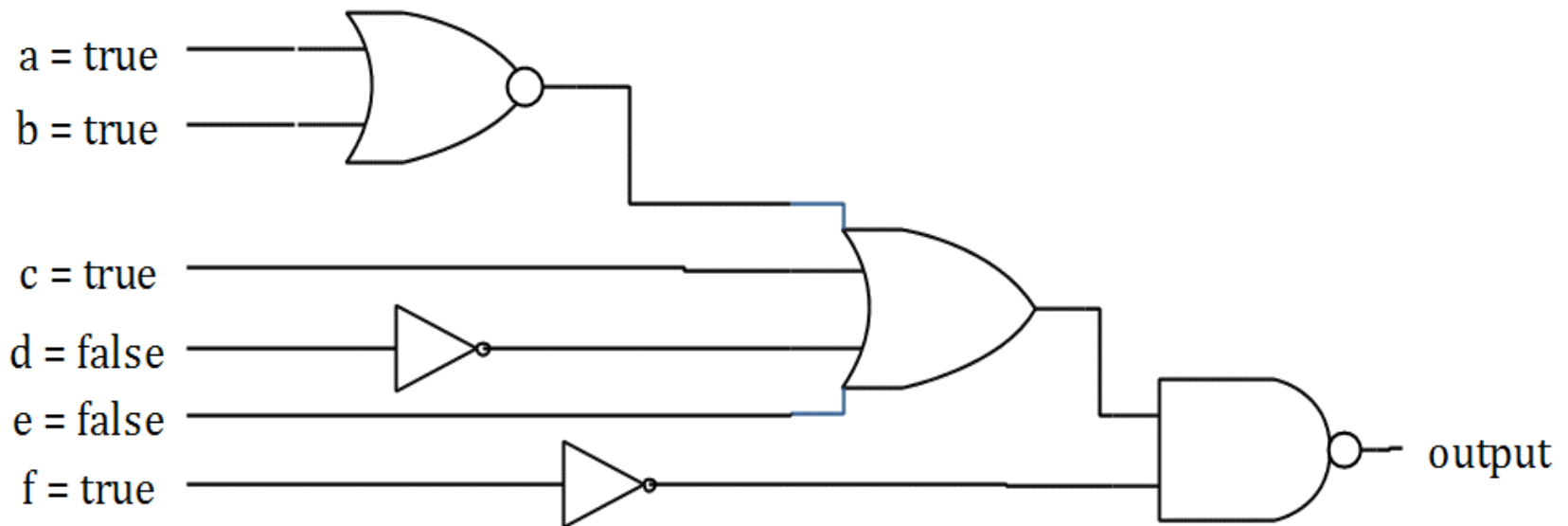


Program Example 02

Problem Statement:

Write a logic circuit solver program in C++ that solves the following logic circuit.

Where all the inputs are logic switches (which can be either Turned ON (True, 1) or Turned OFF (False, 0)). The program should display the output value



Program Example 02

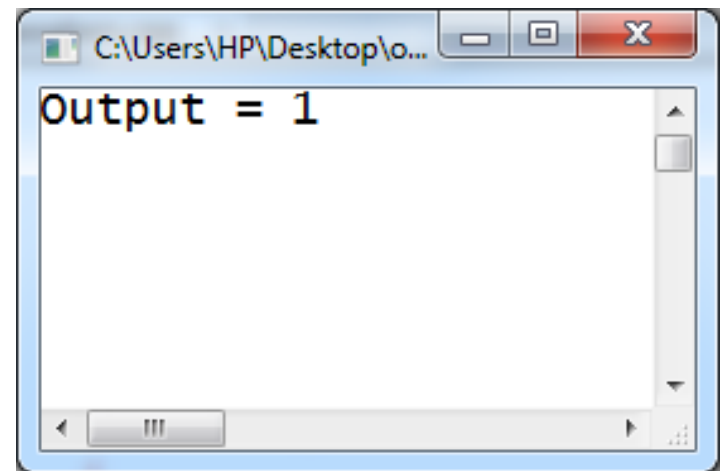
```
#include<iostream>
#include<conio.h>

using namespace std;

int main()
{
    bool a=true, b=true, c=true, d=false, e=false, f=true, output;

    output = (!((a || b) || c || !d || e) && !f);
    cout<<"Output = "<<output;

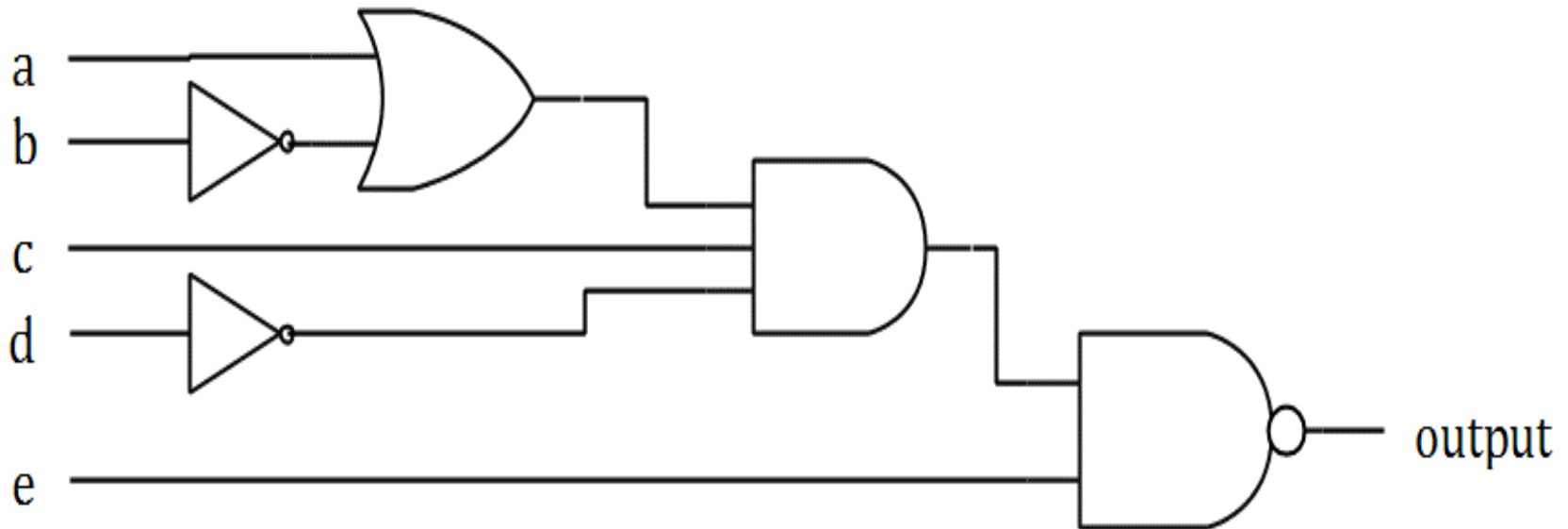
    getch();
    return 0;
}
```



Program Example 03

Problem Statement:

Write a logic circuit solver program in C++ that solves the following logic circuit. Where all the inputs are logic switches (which can be either Turned ON (True, 1) or Turned OFF (False, 0)). The program should display the output value



Program Example 03

```
#include<iostream>
#include<conio.h>

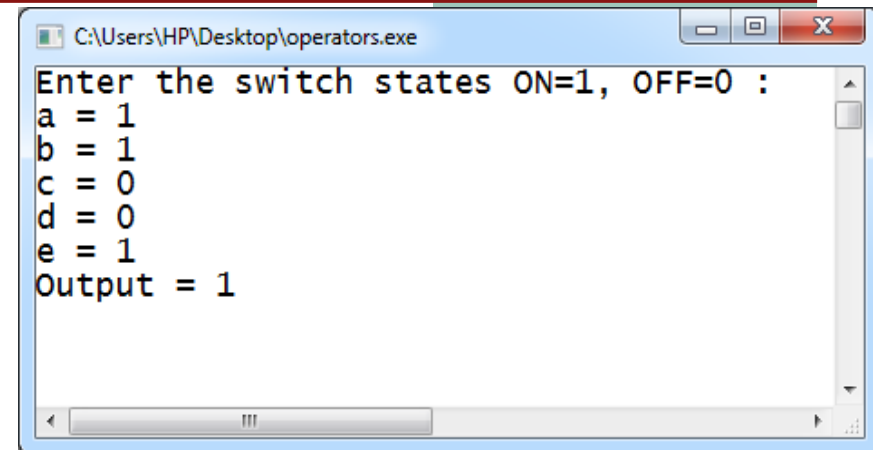
using namespace std;

int main()
{
    bool a, b, c, d, e, output;

    cout<<"Enter the switch states ON=1, OFF=0 : "<<endl;
    cout<<"a = ";    cin>>a;
    cout<<"b = ";    cin>>b;
    cout<<"c = ";    cin>>c;
    cout<<"d = ";    cin>>d;
    cout<<"e = ";    cin>>e;

    output = !(((a || !b) && c && !d) && e);
    cout<<"Output = "<<output;

    getch();
    return 0;
}
```



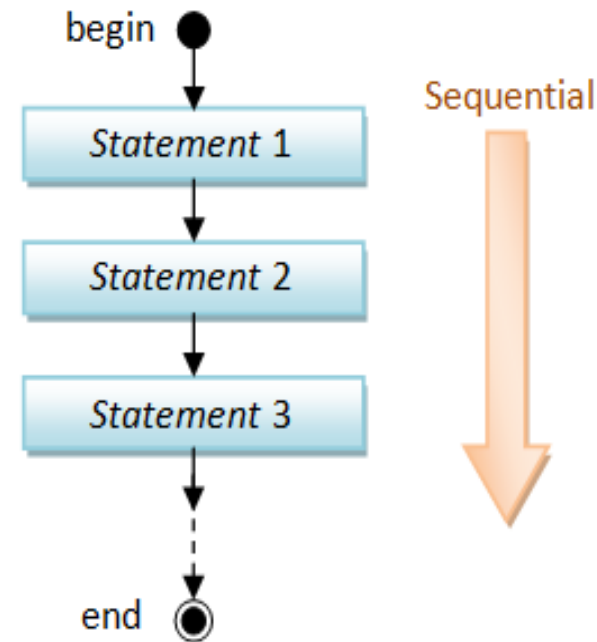
Decision Structures



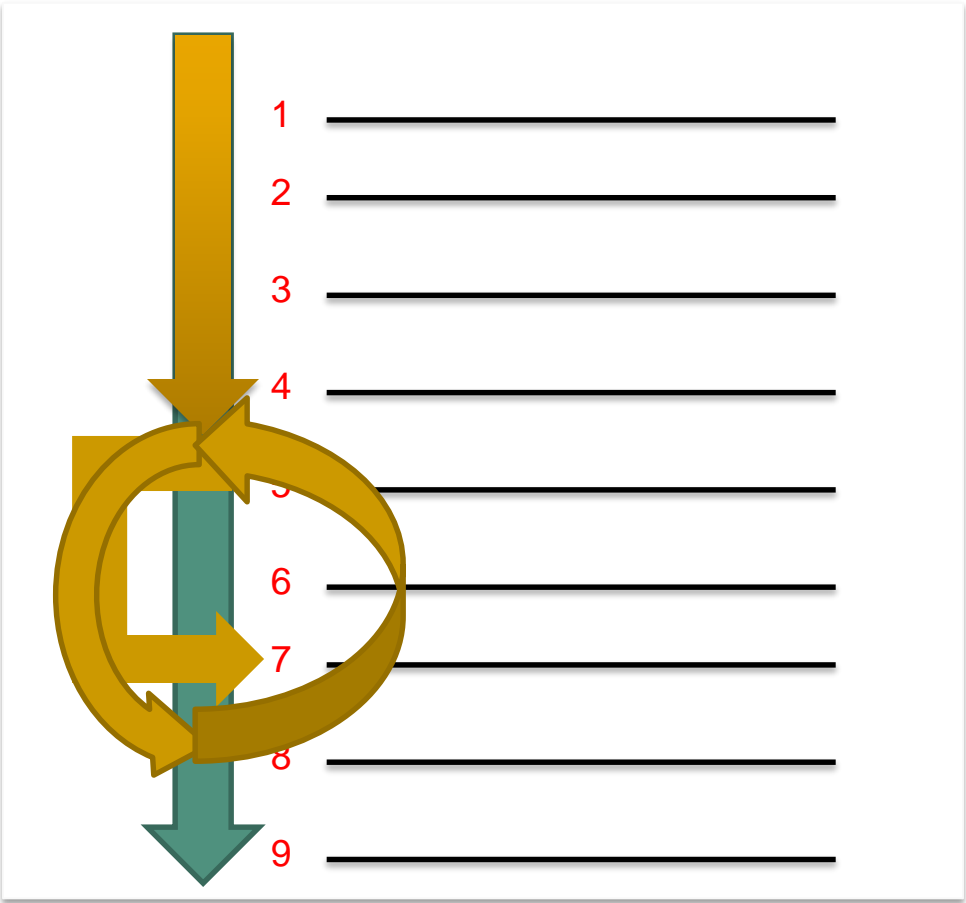
THE IF-ELSE IF STATEMENT AND NESTED
IF STATEMENTS IN C++

Control Statements

- Normally, statements in a program are executed one after the other in the order in which they're written. [sequential execution]
- There are control statements enable you to specify that the next statement to be executed may be other than the next one in sequence. [transfer of control].
- The control statements are categorized in almost two groups:
 - ✓ Selection control statements
 - ✓ Repetition control statements



Transfer of Control



Conditional Control Flow

- Conditional control flow is also referred to as **conditional logic** or **selection logic**.
- It is one of the order in which the program instructions are executed.
- Its execution order differs from the sequential logic.
- It executes the instructions on the basis of one or more conditions.
- If the conditions are satisfied it will execute the instructions else the instructions will not be executed (skipped).

Types of Conditional Control Flow

- In programming we normally have four types of conditional/selection control flow:

On-Way
Selection

Two-Way
Selection

Multi-Way
Selection

Choice-Way
Selection

One-Way Selection

- In one-way selection there is one condition and only one possible choice available either we choose it or not.
- If the condition is satisfied we choose it, and do not choose it, if the condition is not satisfied.
- Like, the boss checks the experience of an employee and adds the bonus, if the experience is more than 2 years.
- Similarly, shopkeeper gives you a discount of 10% if you make the purchase of more than Rs. 5000.
- Both of these examples involve one-way selection.

One-Way Selection

- The syntax of one-way selection is:
 - ✓ The statement is executed if the value of the expression is **true**
 - ✓ The statement is bypassed if the value is **false**; program goes to the next statement
 - ✓ **if** is a reserved word

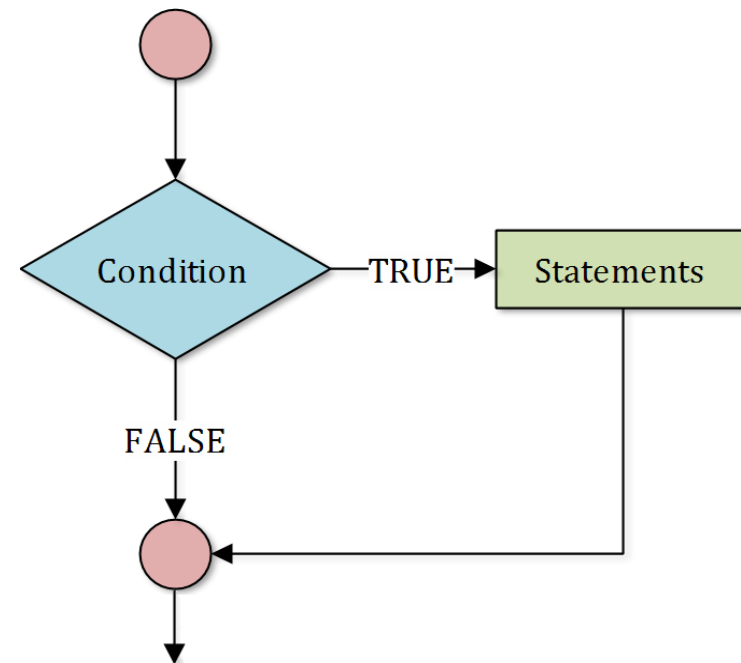
```
if ( Expression )  
    statement ;
```

```
if ( Expression )  
{  
    statement 1 ;  
    statement 2 ;  
    .  
    .  
    statement n ;  
}
```

One-Way Selection

In one-way selection:

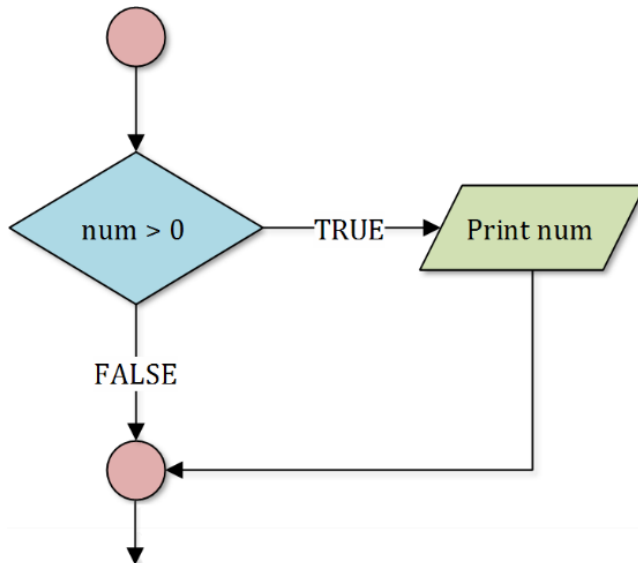
- The statements are executed if the condition is satisfied (true).
- Does nothing when condition is not satisfied (false).



One-Way Selection

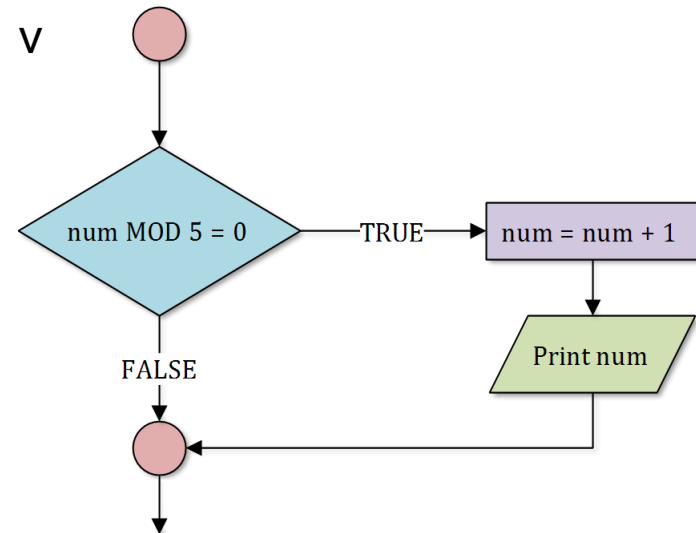
Problem Statement 1:

If the number is positive then display it.



Problem Statement 2:

If the number is multiple of 5 then add 1 to it and display the resultant number.



One-Way Selection (continued)

EXAMPLE 4-9

```
if (score >= 60)
    grade = 'P';
```

In this code, if the expression `(score >= 60)` evaluates to **true**, the assignment statement, `grade = 'P';`, executes. If the expression evaluates to **false**, the statements (if any) following the **if** structure execute. For example, if the value of `score` is 65, the value assigned to the variable `grade` is 'P'.

EXAMPLE 4-10

The following C++ program finds the absolute value of an integer:

//Program: Absolute value of an integer

```
#include <iostream>

using namespace std;

int main()
{
    int number, temp;

    cout << "Line 1: Enter an integer: ";           //Line 1
    cin >> number;                                   //Line 2
    cout << endl;                                    //Line 3

    temp = number;                                   //Line 4

    if (number < 0)                                   //Line 5
        number = -number;                           //Line 6

    cout << "Line 7: The absolute value of "
         << temp << " is " << number << endl;      //Line 7

    return 0;
}
```

Sample Run: In this sample run, the user input is shaded.

Line 1: Enter an integer: -6734

Line 7: The absolute value of -6734 is 6734

One-Way Selection (continued)

EXAMPLE 4-11

Consider the following statement:

```
if score >= 60      //syntax error
    grade = 'P';
```

This statement illustrates an incorrect version of an **if** statement. The parentheses around the logical expression are missing, which is a syntax error.

EXAMPLE 4-12

Consider the following C++ statements:

```
if (score >= 60);    //Line 1
    grade = 'P';     //Line 2
```

Because there is a semicolon at the end of the expression (see Line 1), the **if** statement in Line 1 terminates. The action of this **if** statement is null, and the statement in Line 2 is not part of the **if** statement in Line 1. Hence, the statement in Line 2 executes regardless of how the **if** statement evaluates.

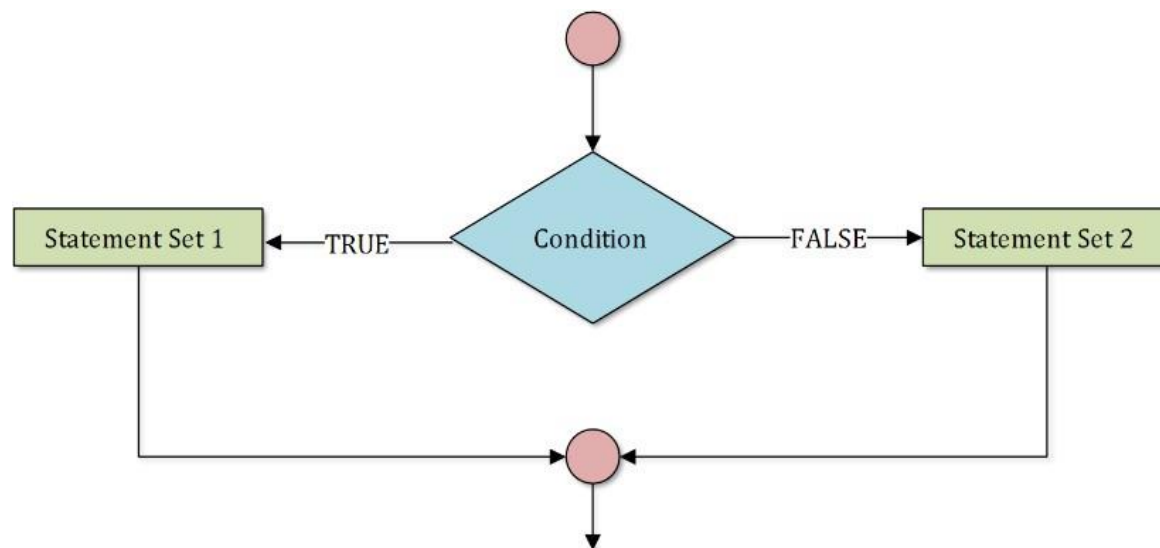
Two-Way Selection

- In two-way selection there is one condition and two possible choices available either we choose first one or the second one.
- If the condition is satisfied we choose first choice and the second choice if the condition is not satisfied.
- Like, the teacher checks the roll number of a student, if it is even he/she is from section 2 otherwise he/she is from section 1.
- It is an example involving two-way selection.

Two-Way Selection (if-else condition)

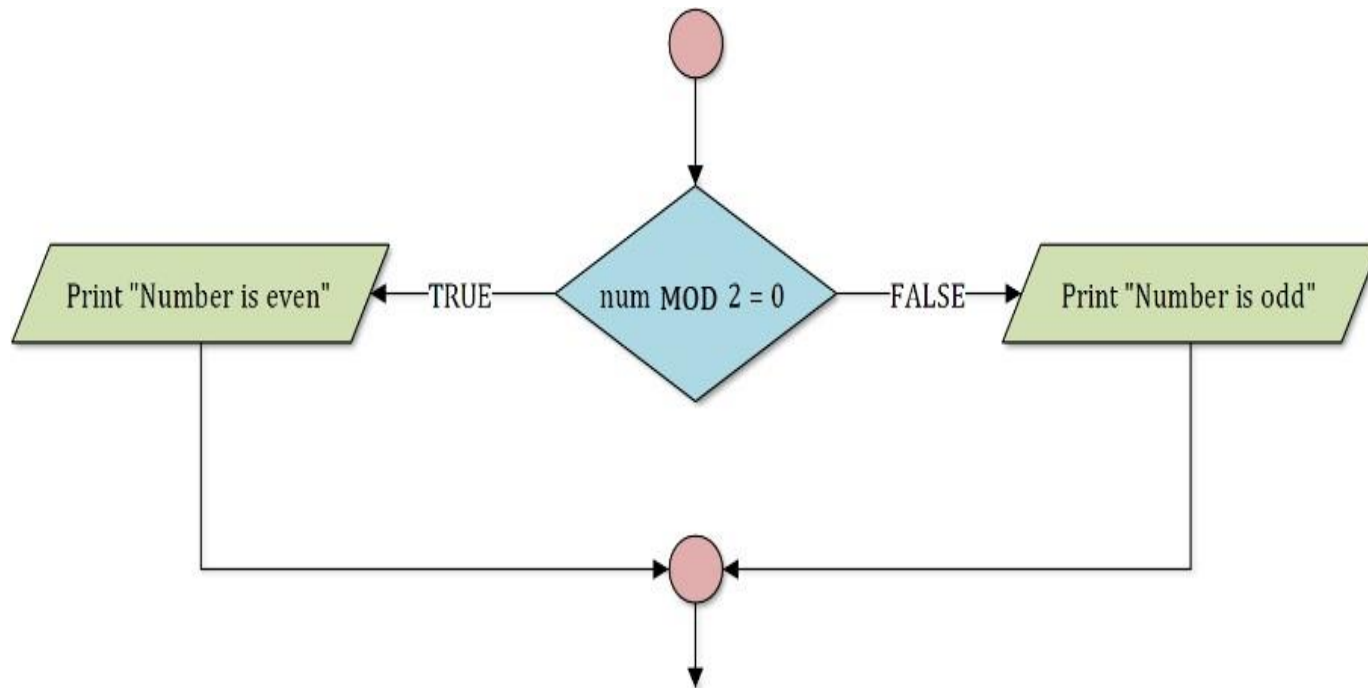
In two-way selection:

- If the condition is satisfied (true), the 1st set of statements is executed.
- If the condition is not satisfied (false), the 2nd set of statements is executed.



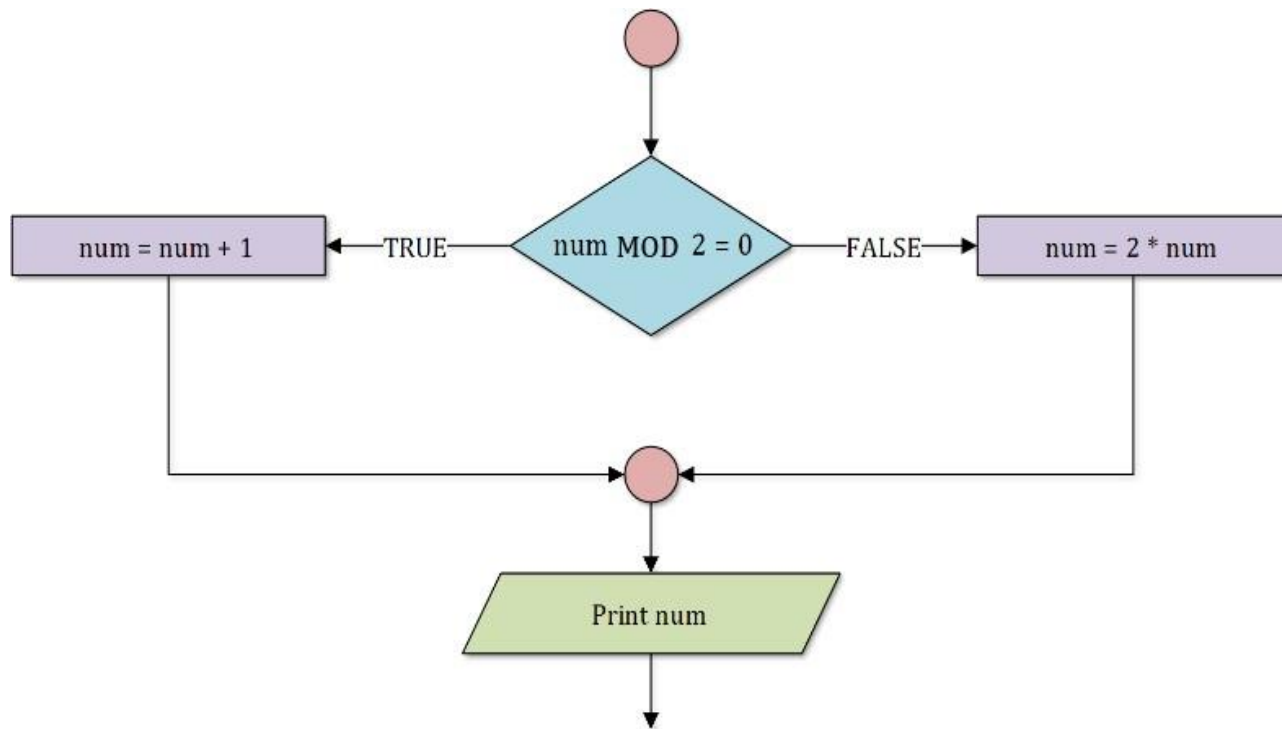
Two-Way Selection - Example

- **Problem Statement 1:** If the number is even, display “Number is even” else display “Number is odd”.



Two-Way Selection - Example

- **Problem Statement 2:** If the number is even, make it odd else double it and finally display the number.



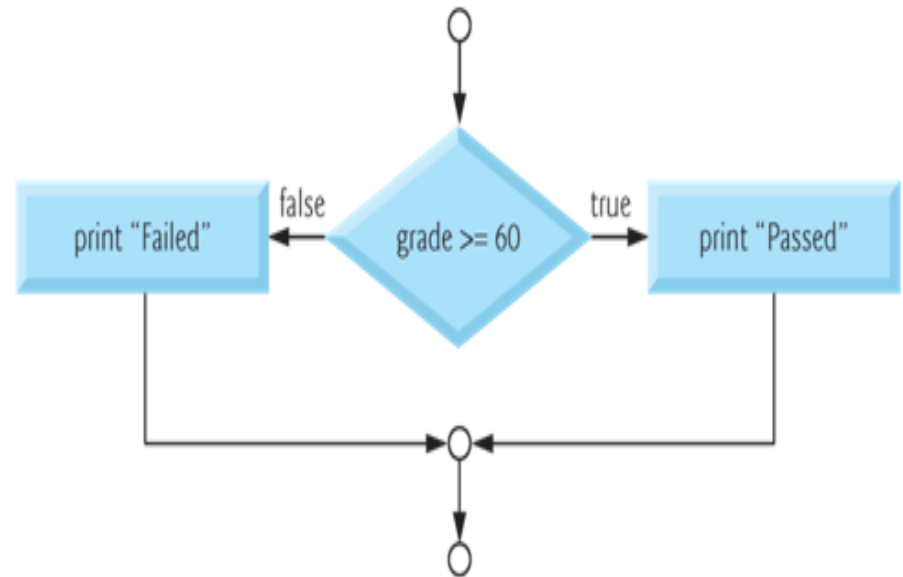
In C++ , The syntax for the If...Else statement

```
if ( Expression)
    statement1 ;
else
    statement2 ;
```

```
if ( Expression)
{
    statements 1 ;
    .
    statement n ;
}
else
{
    statements 1 ;
    .
    statement n ;
}
```

```
if ( grade >= 60 )
    cout <<"Passed\n";
else
    cout <<"Failed\n"
```

- If expression is **true**, statement1 is executed; otherwise, statement2 is executed
- statement1 and statement2 are any C++ statements
- **else** is a reserved word



Two-Way Selection (continued)

EXAMPLE 4-13

Consider the following statements:

```
if (hours > 40.0)           //Line 1
    wages = 40.0 * rate +
        1.5 * rate * (hours - 40.0); //Line 2
else                         //Line 3
    wages = hours * rate;    //Line 4
```

If the value of the variable `hours` is greater than 40.0, then the `wages` include overtime payment. Suppose that `hours` is 50. The expression in the `if` statement, in Line 1, evaluates to `true`, so the statement in Line 2 executes. On the other hand, if `hours` is 30, or any number less than or equal to 40, the expression in the `if` statement, in Line 1, evaluates to `false`. In this case, the program skips the statement in Line 2 and executes the statement in Line 4—that is, the statement following the reserved word `else` executes.

Two-Way Selection (continued)

EXAMPLE 4-14

The following statements show an example of a syntax error:

```
if (hours > 40.0); //Line 1
    wages = 40.0 * rate +
        1.5 * rate * (hours - 40.0); //Line 2
else //Line 3
    wages = hours * rate; //Line 4
```

The semicolon at the end of the **if** statement (see Line 1) ends the **if** statement, so the statement in Line 2 separates the **else** clause from the **if** statement. That is, **else** is all by itself. Because there is no stand-alone **else** statement in C++, this code generates a syntax error.

Example

```
if (age > 18)
{
    cout << "Eligible to vote." << endl;
    cout << "No longer a minor." << endl;
}
else
{
    cout << "Not eligible to vote." << endl;
    cout << "Still a minor." << endl;
}
```