Simple if statement in C

Simple if Syntax

The form of an if statement is as follows:

```
if(condition)
   statement;
```

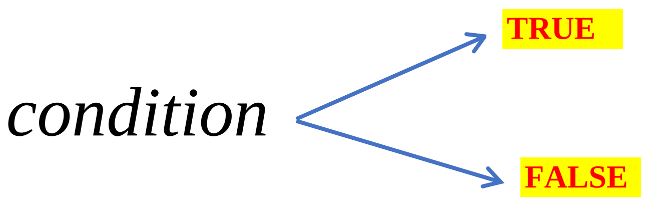
← No semicolon after if

- If the *condition* is true, immediate statement following **if** is executed
- If the *condition* is false, the *statement after* ***if*** *is not executed*

If multiple statements are to be executed after **if**, we must include them in curly braces

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

- If the *condition* is true, block of statements (called compound statement) inside if is executed
- If the *condition* is false, block of statements (called compound statement) inside if is not executed



In C/C++,

1. ZERO → represents FALSE condition
2. Non-zero → represents TRUE condition

   Examples of non-zero values → 5, -5.1, 100, -206 etc if(5)

↑

5 is non-zero and represents **TRUE**

- **> (greater than)** is relational operator. Note that relational operators return either true or false

Operator	Meaning	Example
= =	Equality	5 == 5 // returns True
!=	Not Equal to	5 != 5 // returns False
<	Less Than	5 < 5.5 // returns True
<=	Less Than or Equal	5 <= 5 // returns True
>	Greater Than	5 > 5.5 // returns True
>=	Greater Than or Equal	6.3 >= 5 // returns True
Relational operators		

Note that every operator in C++ must return some value. For example, + operator returns sum of two numbers, * operator return multiplication of two numbers etc.

Practice Programs

```
(i)
#include<stdio.h>
void main()
{
        system("color fc");
        // e = = light yellow = Output window background color
        //c = = Light red = Output window text color
        int x = 5;

        if(x)
        printf("EngineersTutor.com")
}
```

```
(ii)
#include<stdio.h>
void main()
{
        system("color ec");
        // e = = light yellow = Output window background color
        //c = = Light red = Output window text color

        int x = 5;

        if(x>10)
        printf("EngineersTutor.com");
}
```

(iii)

```
#include<stdio.h>
void main()
{
        system("color ec");
        // e = = light yellow = Output window background color
        //c = = Light red = Output window text color

        int x = 5;

        if(x == 10)
        printf("EngineersTutor.com");
}
```

(iv)

```
#include<stdio.h>
int main()
{
        system("color ec");
        // e = = light yellow = Output window background color
        //c = = Light red = Output window text color

        int x = 5, y = 10;

        if(x+y)
        printf("EngineersTutor.com");
}
```

(v)

```
#include<stdio.h>
void main()
{
        system("color ec");
        // e = = light yellow = Output window background color
        //c = = Light red = Output window text color

        int x = 5, y = 10;

        if( (x+y)>30 )
        printf("EngineersTutor.com");
}
```

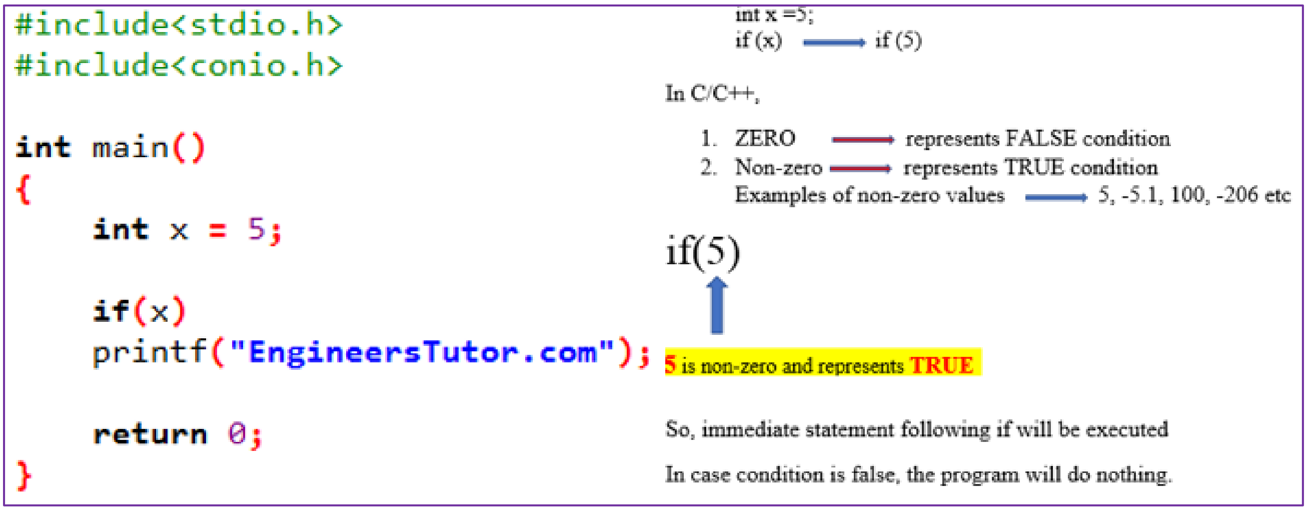
```
(vi)
#include<stdio.h>

void main()
{
        system("color ec");
        // e = = light yellow = Output window background color
        //c = = Light red = Output window text color

        int x = 5, y = 10;

        if( (x+y)>30 )
        {
                printf("EngineersTutor.com");
                printf("Teach Easy");
        }
}
```

```
(vii)
#include<stdio.h>
void main()
{
        system("color ec");
        // e = = light yellow = Output window background color
        //c = = Light red = Output window text color

        int x = 5, y = 10;

        if( (x+y)<30 )
                printf("EngineersTutor.com\n");
                printf("Teach Easy\n");
           printf("Albert\n");
           printf("Stephen");
}
```

```
(vii) Testing for Leap year
#include<stdio.h>

void main()
{
        int year;
        printf("enter year \n");
        scanf("%d", &year);

        if((year%400==0)||((year%4==0)&&(year%100!=0)))
                printf("given year is leap year \n");
        else
                printf("not leap year \n");
}
```

Program explanation

Program 1



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

int main()
{
    int x = 5;

    if(x)
    printf("EngineersTutor.com");

    return 0;
}
```

int x =5;
if (x) → if (5)

In C/C++,

1. ZERO → represents FALSE condition
2. Non-zero → represents TRUE condition
   Examples of non-zero values → 5, -5.1, 100, -206 etc

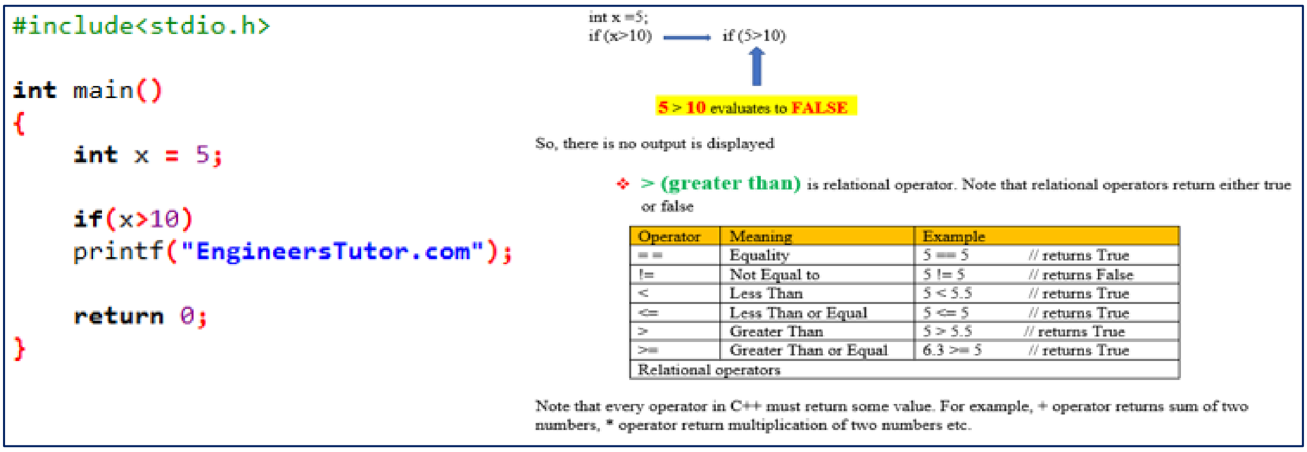
if(5)

5 is non-zero and represents TRUE

So, immediate statement following if will be executed

In case condition is false, the program will do nothing.

Program 2



```
#include<stdio.h>

int main()
{
    int x = 5;

    if(x>10)
    printf("EngineersTutor.com");

    return 0;
}
```

int x =5;
if (x>10) → if (5>10)

5 > 10 evaluates to FALSE

So, there is no output is displayed

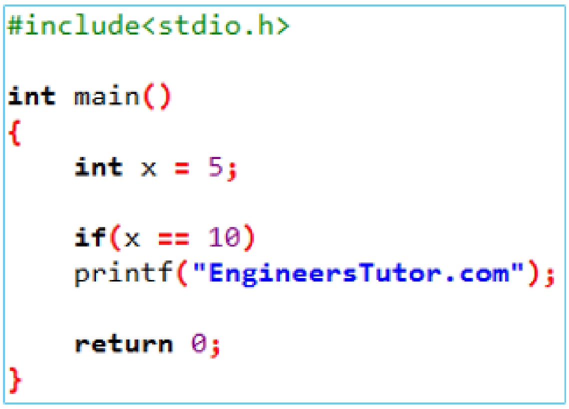
- > (greater than) is relational operator. Note that relational operators return either true or false

Operator	Meaning	Example
= =	Equality	5 == 5 // returns True
!=	Not Equal to	5 != 5 // returns False
<	Less Than	5 < 5.5 // returns True
<=	Less Than or Equal	5 <= 5 // returns True
>	Greater Than	5 > 5.5 // returns True
>=	Greater Than or Equal	6.3 >= 5 // returns True

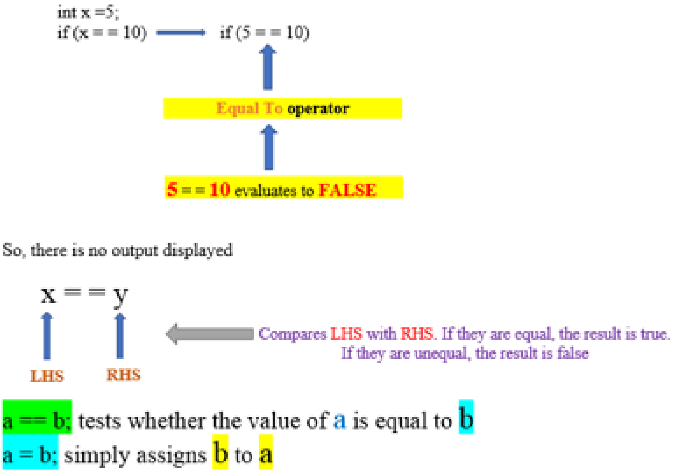
Relational operators

Note that every operator in C++ must return some value. For example, + operator returns sum of two numbers, * operator return multiplication of two numbers etc.

Program 3



```
#include<stdio.h>

int main()
{
    int x = 5;

    if(x == 10)
    printf("EngineersTutor.com");

    return 0;
}
```



int x =5;
if (x = = 10) → if (5 = = 10)

Equal To operator

5 = = 10 evaluates to FALSE

So, there is no output displayed

x = = y

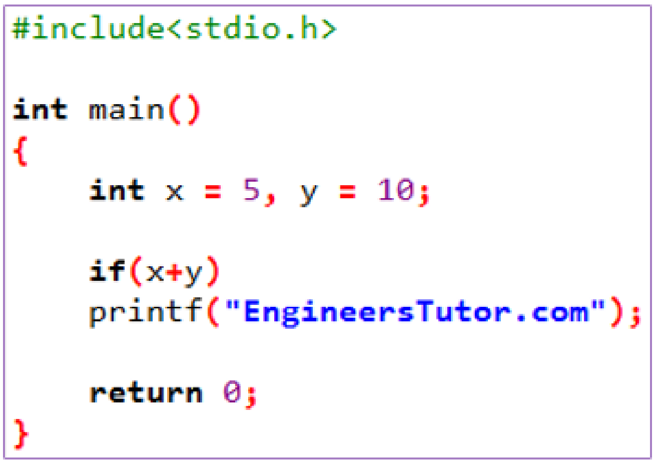
LHS RHS

Compares LHS with RHS. If they are equal, the result is true. If they are unequal, the result is false

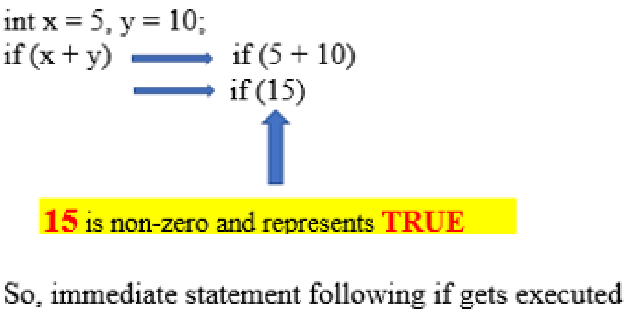
a == b; tests whether the value of a is equal to b

a = b; simply assigns b to a

Program 4



```
#include<stdio.h>

int main()
{
    int x = 5, y = 10;

    if(x+y)
    printf("EngineersTutor.com");

    return 0;
}
```

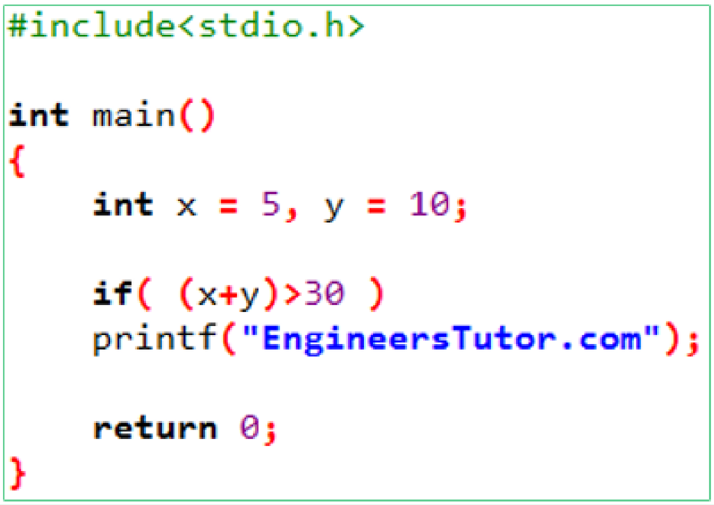


int x = 5, y = 10;
if (x + y) → if (5 + 10)
→ if (15)

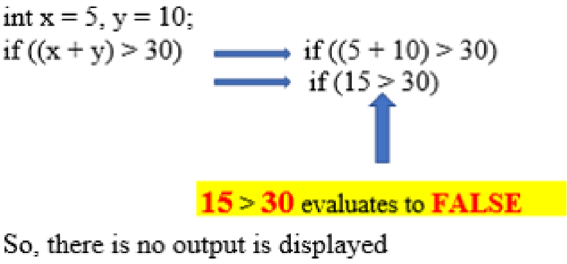
15 is non-zero and represents TRUE

So, immediate statement following if gets executed

Program 5



```
#include<stdio.h>

int main()
{
    int x = 5, y = 10;

    if( (x+y)>30 )
    printf("EngineersTutor.com");

    return 0;
}
```

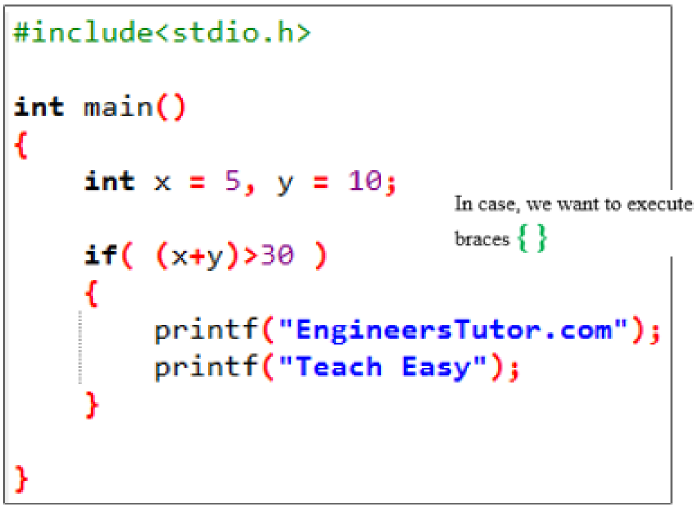


int x = 5, y = 10;
if ((x + y) > 30) → if ((5 + 10) > 30)
→ if (15 > 30)

15 > 30 evaluates to FALSE

So, there is no output is displayed

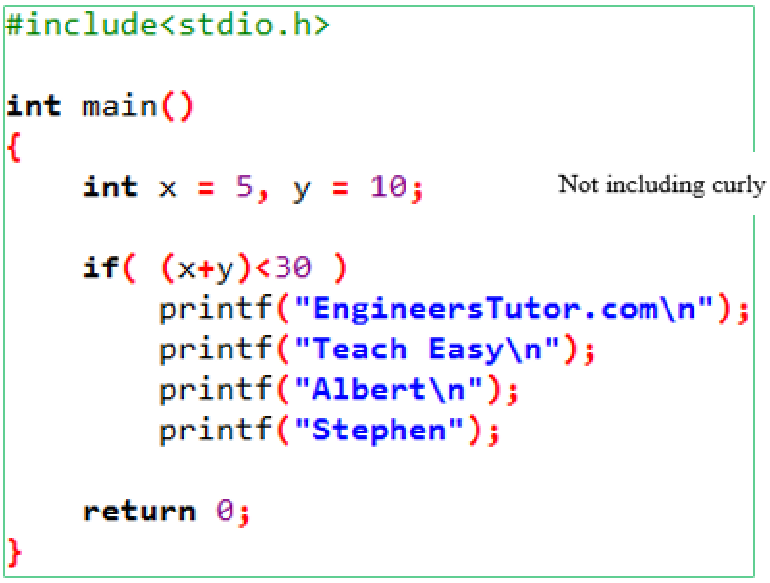
Program 6



```
#include<stdio.h>

int main()
{
    int x = 5, y = 10;

    if( (x+y)>30 )
    {
        printf("EngineersTutor.com");
        printf("Teach Easy");
    }

}
```

In case, we want to execute more than one statement after if, we must include them within curly braces { }

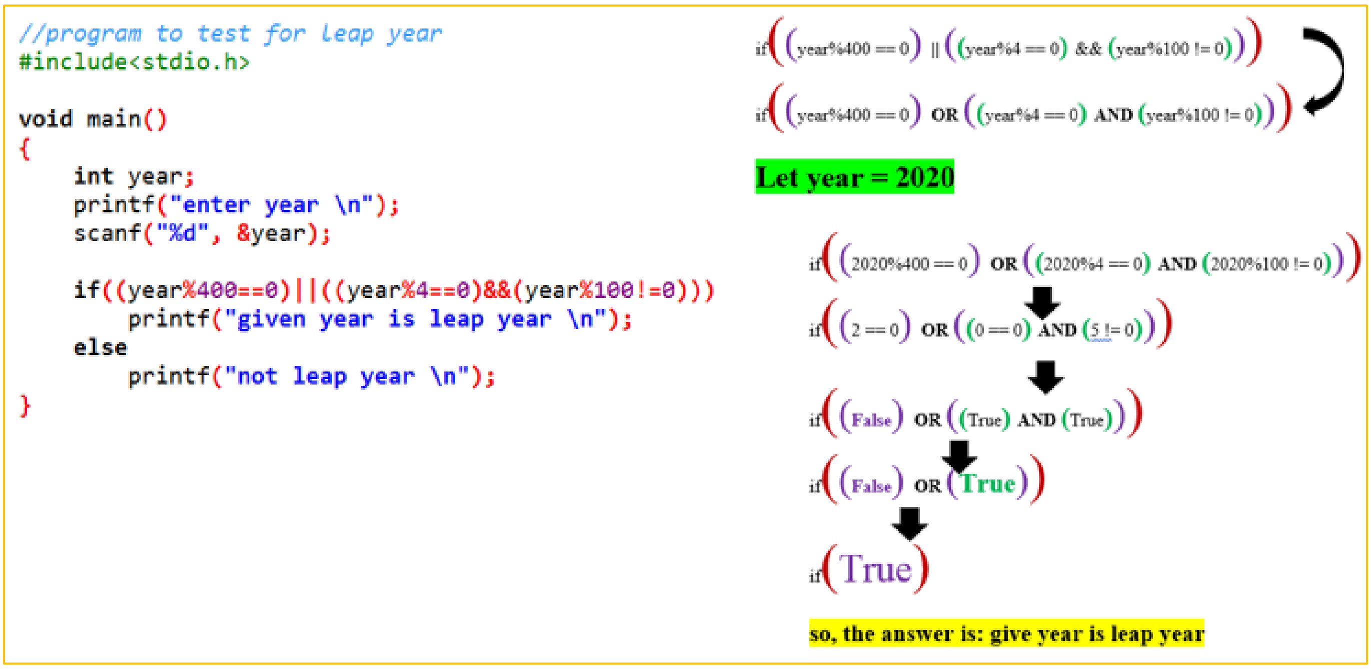
Program 7



```
#include<stdio.h>

int main()
{
    int x = 5, y = 10;

    if( (x+y)<30 )
        printf("EngineersTutor.com\n");
        printf("Teach Easy\n");
        printf("Albert\n");
        printf("Stephen");

    return 0;
}
```

Not including curly braces { } **will create confusion**

Program 8



```
//program to test for Leap year
#include<stdio.h>

void main()
{
    int year;
    printf("enter year \n");
    scanf("%d", &year);

    if((year%400==0)||((year%4==0)&&(year%100!=0)))
        printf("given year is leap year \n");
    else
        printf("not leap year \n");
}
```

if((year%400 == 0) || ((year%4 == 0) && (year%100 != 0)))

if((year%400 == 0) OR ((year%4 == 0) AND (year%100 != 0)))

Let year = 2020

if((2020%400 == 0) OR ((2020%4 == 0) AND (2020%100 != 0)))

if((2 == 0) OR ((0 == 0) AND (5 != 0)))

if((False) OR ((True) AND (True)))

if((False) OR (True))

if(True)

so, the answer is: give year is leap year