

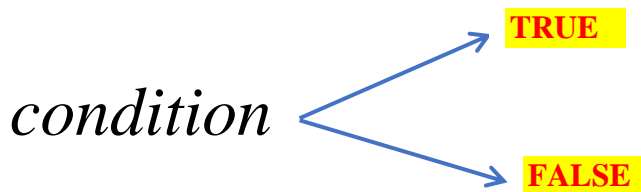
Simple if - else statement in C

Simple if - else Syntax

The form of an if statement is as follows:

```
if(condition)    ← No semicolon after if
    statement;
else              ← No semicolon after else
    statement;
```

- If the *condition* is true, immediate statement following **if** is executed
- If the *condition* is false, immediate statement following **else** is 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()
{
    int x = 5;

    if(x)
        printf("Stephen Hawking");
    else
        printf("Albert Einstein");
}
```

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

void main()
{
    int x = 5;

    if(x>10)
        printf("Stephen Hawking");
    else
        printf("Albert Einstein");
}
```

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

void main()
{
    int x = 5;

    if(x == 10)
        printf("Stephen Hawking");
    else
        printf("Albert Einstein");
}
```

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

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

    if(x+y)
        printf("Stephen Hawking");
    else
        printf("Albert Einstein");
}
```

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

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

    if( (x+y)>30 )
        printf("Stephen Hawking");
    else
        printf("Albert Einstein");
}
```

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

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

    if( (x+y)>30 )
    {
        printf("Stephen Hawking\n");
        printf("Cosmology");
    }

    else
    {
        printf("Albert Einstein\n");
        printf("Physics");
    }
}
```

```

(vii)
#include<stdio.h>

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

    if( (x+y)>30 )
        printf("Stephen Hawking\n");
        printf("Cosmology");

    else
        printf("Albert Einstein\n");
        printf("Physics");
}

```

```

(viii) 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 explanations

Program 1

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

```
int main()
{
    int x = 5;

    if(x)
        printf("Stephen Hawking");
    else
        printf("Albert Einstein");

    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 statement following **else** will be executed

Program 2

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

```
int main()
{
    int x = 5;

    if(x>10)
        printf("Stephen Hawking");
    else
        printf("Albert Einstein");

    return 0;
}
```

int x=5;
if (x>10) → if (5>10)

5 > 10 evaluates to FALSE

So, Albert Einstein 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.

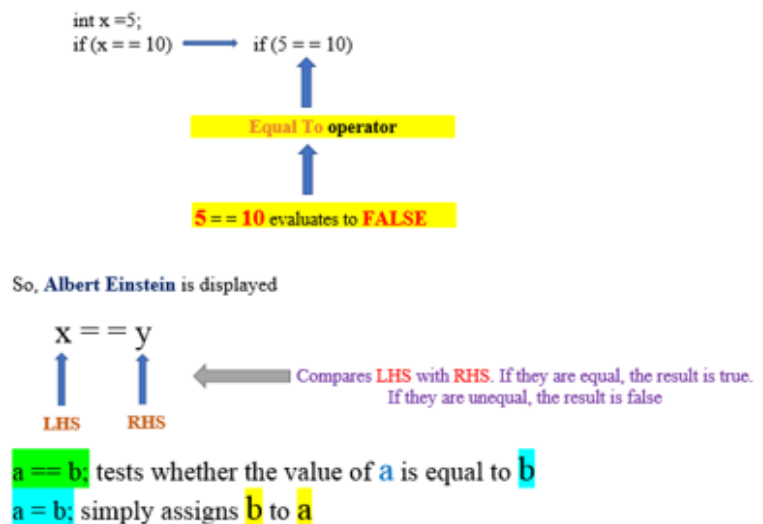
Program 3

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

```
int main()
{
    int x = 5;

    if(x == 10)
        printf("Stephen Hawking");
    else
        printf("Albert Einstein");

    return 0;
}
```



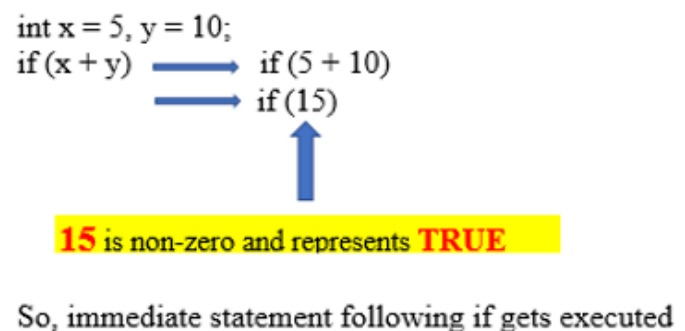
Program 4

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

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

    if(x+y)
        printf("Stephen Hawking");
    else
        printf("Albert Einstein");

    return 0;
}
```



Program 5

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

```
int main()
```

```
{
```

```
    int x = 5, y = 10;
```

```
    if( (x+y)>30 )
```

```
        printf("Stephen Hawking");
```

```
    else
```

```
        printf("Albert Einstein");
```

```
    return 0;
```

```
}
```

int x = 5, y = 10;

if ((x + y) > 30)

→ if ((5 + 10) > 30)

→ if (15 > 30)



15 > 30 evaluates to **FALSE**

So, Albert Einstein is displayed

Program 6

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

```
int main()
```

```
{
```

```
    int x = 5, y = 10;
```

```
    if( (x+y)>30 )
```

```
    {
```

```
        printf("Stephen Hawking\n");
```

```
        printf("Cosmology");
```

```
    }
```

```
    else
```

```
    {
```

```
        printf("Albert Einstein\n");
```

```
        printf("Physics");
```

```
    }
```

```
    return 0;
```

```
}
```

In case, we want to execute more than one statement

after **if** or **else** we must include them within curly braces { }

Program 7

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

```
int main()
```

```
{
```

```
    int x = 5, y = 10;
```

```
    if( (x+y)>30 )
```

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

```
        printf("Stephen Hawking\n");
```

```
        printf("Cosmology");
```

```
    else
```

```
        printf("Albert Einstein\n");
```

```
        printf("Physics");
```

```
    return 0;
```

```
}
```

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