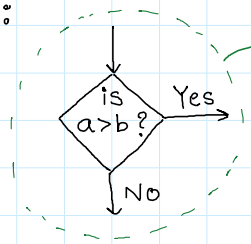
Recall :



→ In code is called a conditional statement.

Example : I/P : a, b

O/P : $\begin{cases} a & \text{if } a > b \\ b & \text{otherwise} \end{cases}$

Conditionals like these are solved using if statements.

```
if (~~~~) {
    ~~~~
    ~~~~
    ~~~~
}
```

If the condition within ( ) is true, then execute the entire code block within { }.

Thus,

```
if ( a > b) {
    cout << a << endl;
}
if ( a <=b) {
    cout << b << endl;
}
```

In the above example, instead of checking again for b > a in the second if block, we know that if the first if condition does not get fulfilled, then the second block must be executed no matter what. This can be achieved using an if - else block.

If this is fulfilled, else block won't execute

```
if ( a > b) {
    cout << a << endl;
}
else {
    cout << b << endl;
}
```

If if block doesn't execute, then this else block will execute.

NEW CONCEPT :

`cin >> n;` → Waits for user to give input and assigns it to n at its address.

Example- `int a;`
`cin >> a;`

① variable a will get initialized with a random, 'garbage' value.

② Program waits for user to give input (an integer) to a and stores a with that value.

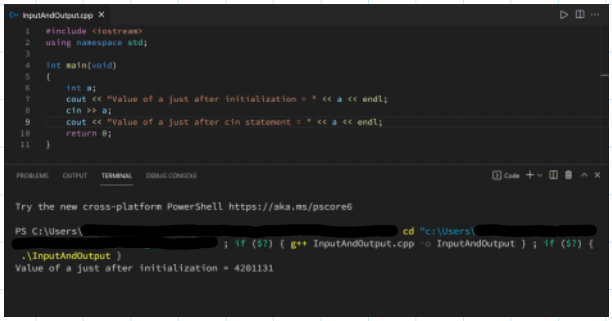


Fig ①

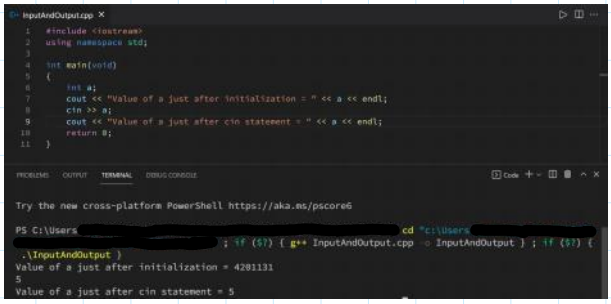
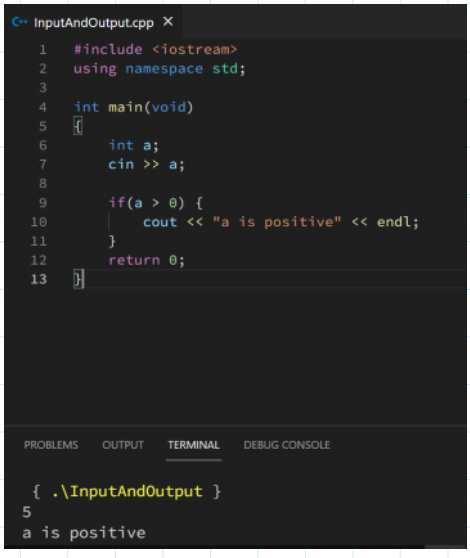


Fig ②

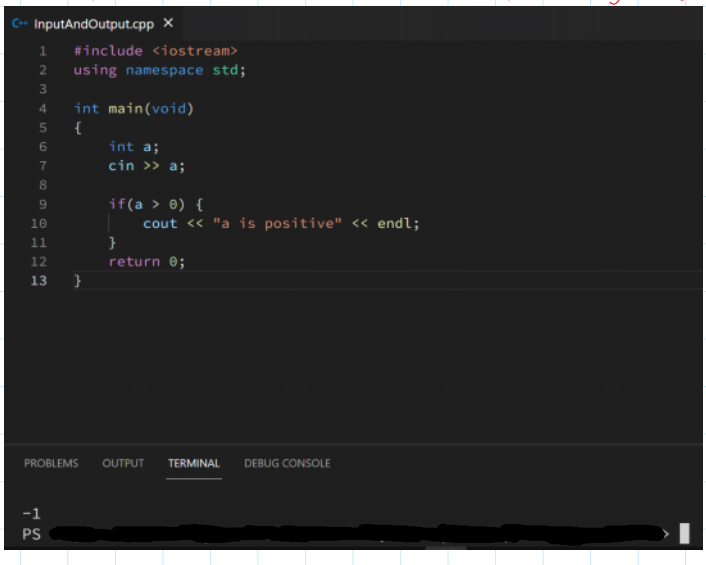
Back to if-else :

Here, the if block doesn't execute and the program exits without printing anything

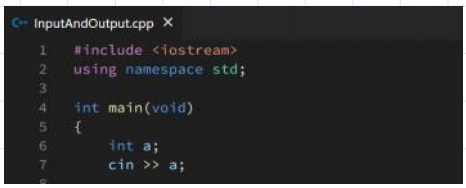
①



②



③



④

```cpp
int main(void)
{
    int a;
    cin >> a;

    if(a > 0) {
        cout << "a is positive" << endl;
    }
    else {
        cout << "a is not positive" << endl;
    }
    return 0;
}
```

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

 .\InputAndOutput }
5
a is positive
```

```cpp
int main(void)
{
    int a;
    cin >> a;

    if(a > 0) {
        cout << "a is positive" << endl;
    }
    else {
        cout << "a is not positive" << endl;
    }
    return 0;
}
```

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

 .\InputAndOutput }
-2
a is not positive
```

Program to compare two numbers :

InputAndOutput.cpp

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

int main()
{
    int a, b;
    cout << "Enter the value of a: ";
    cin >> a;
    cout << "Enter the value of b: ";
    cin >> b;

    if(a > b) {
        cout << "a is greater than b" << endl;
    }
    if (b > a) {
        cout << "b is greater than a" << endl;
    }
    return 0;
}
```

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

 .\InputAndOutput }
Enter the value of a: 6
Enter the value of b: 4
a is greater than b
```

InputAndOutput.cpp

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

int main()
{
    int a, b;
    cout << "Enter the value of a: ";
    cin >> a;
    cout << "Enter the value of b: ";
    cin >> b;

    if(a > b) {
        cout << "a is greater than b" << endl;
    }
    if (b > a) {
        cout << "b is greater than a" << endl;
    }
    return 0;
}
```

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

 .\InputAndOutput }
Enter the value of a: 3
Enter the value of b: 6
b is greater than a
```

Note : cin ignores ENTER (\n), TAB (\t) and SPACE ( ) while taking input. These are called whitespace characters. Use cin.get() to read these whitespace characters.

Using nested if - else :

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

int main()
{
    int a;
    cin >> a;

    if(a > 0) {
        cout << "a is positive" << endl;
    }
    else {
        cout << "a is not positive" << endl;
    }
    return 0;
}
```

→ In this else block, we can see that either a will be negative or 0. So we can further break down this else block.

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

int main()
{
    int a;
    cin >> a;
    if(a > 0) {
        cout << "a is positive" << endl;
    }
    else {
        if (a == 0) {
            cout << "a is 0" << endl;
        }
        else {
            cout << "a is negative" << endl;
        }
    }
    return 0;
}
```

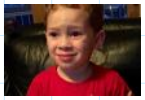
```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE
 .\InputAndOutput }
5
a is positive
```

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

int main()
{
    int a;
    cin >> a;
    if(a > 0) {
        cout << "a is positive" << endl;
    }
    else {
        if (a == 0) {
            cout << "a is 0" << endl;
        }
        else {
            cout << "a is negative" << endl;
        }
    }
    return 0;
}
```

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE
 .\InputAndOutput }
0
a is 0
```

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

int main()
{
    int a;
    cin >> a;
    if(a > 0) {
        cout << "a is positive" << endl;
    }
    else {
        if (a == 0) {
            cout << "a is 0" << endl;
        }
        else {
            cout << "a is negative" << endl;
        }
    }
    return 0;
}
```

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE
 .\InputAndOutput }
-9
a is negative
```



Code ganda hote jaa raha hai. Ek-do baar aur nested if-else kar diya to coding chorni padhegi.

Solution : else if

```
if ( ~~~~ ) {

}
else if ( ~~~~ ) {

}
else {

}
```

```cpp
int a ;
cout<<" enter the value of a "<<endl;
cin>>a;


if(a>0) {
    cout<<" A is positive"<< endl;
}
else if(a<0) {
    cout<<" A is negative"<<endl;
}
else {
    cout<<" A is 0"<<endl;
}
```

Note : else if and else are optional.
else can be used as a default case.

Homework : Output ??

(1)

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

int main() {

        int a = 9;
        if(a == 9) {
                cout<< "NINEY";
        }

        if(a > 0){
                cout<<"POSITIVE";
        }
        else
        {
                cout<<"NEGATIVE";
        }
}
```

Answer : NINEYPOSITIVE

(2)

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

int main() {

        int a = 2;
        int b = a+1;

        if((a=3)==b) {
                cout<<a;
        }
        else
        {
                cout<<a+1;
        }
}
```

Answer : 3

a is assigned 3.
Now, a == b is true.

(3)

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

int main() {

        int a = 24;

        if(a > 20){
                cout<< "Love ";
        }
        else if(a == 24) {
                cout<<"Lovely";
        }
        else
        {
                cout<<"Babbar";
        }
        cout<<a;
}
```

Answer : Love24

(4) Code Homework :

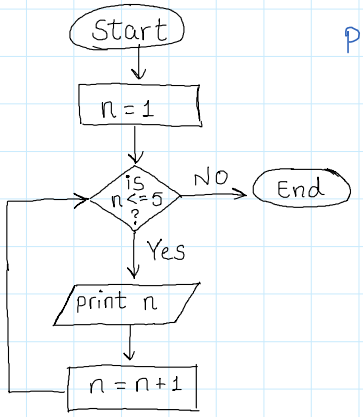
```cpp
#include <iostream>
using namespace std;

int main()
{
    char a;
    cin >> a;
    // 'A' is 65
    // 'a' is 97
    // '0' is 48
    if(a >= 'A' && a <= 'Z') {
        cout << "This is upper case" << endl;
    }
    else if(a >= 'a' && a <= 'z') {
        cout << "This is lower case" << endl;
    }
    else if(a >= '0' && a <= '9') {
        cout << "This is a digit" << endl;
    }
    return 0;
}
```

We can also use ASCII values

Loops

07 January 2022 20:16

Recollect from flowcharts :



Print from 1 to 5.

while loop :

```
while (condition) {
}
```

जब तक ये तुरू हैं

तब तक ये करते रहो

while the condition is true, keep on executing the block

Example :

Print 1 to N.

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

int main()
{
    int n;
    cin >> n;
    int i = 1;
    while(i <= n) {
        cout << i << " ";
        i = i + 1;
    }
    return 0;
}
```

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE
.\InputAndOutput }
8
1 2 3 4 5 6 7 8
```

Example : Find sum from 1 to n.

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

int main()
{
    int n;
    cin >> n;
    int sum = 0;
    int i = 1;
    while(i <= n) {
        sum = sum + i;
        i = i + 1;
    }
    cout << "Sum from " << 1 << " to " << n << " = " << sum << endl;
    return 0;
}
```

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE
 { .\InputAndOutput }
8
Sum from 1 to 8 = 36
```

Example: Find sum of all even numbers from 1 to n.

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

int main()
{
    int n;
    cin >> n;
    int sum = 0;
    int i = 2;
    while(i <= n) {
        if(i % 2 == 0)
            sum = sum + i;
        i = i + 1;
    }
    cout << "Sum from " << 1 << " to " << n << " = " << sum << endl;
    return 0;
}
```

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE
8
Sum from 1 to 8 = 20
```

Homework: Fahrenheit to Celsius.

$$C = \frac{5}{9}(F - 32)$$

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

int main()
{
    float fahrenheit;
    cin >> fahrenheit;
    float celsius = (5.0/9) * (fahrenheit - 32);
    cout << fahrenheit << " F = " << celsius << " C" << endl;

    return 0;
}
```

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE
 { .\InputAndOutput }
100
100 F = 37.7778 C
```

Example: Prime or not ?

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

int main()
{
    int n;
    cin >> n;
    int i = 2;
    while(i < n) {
        if(n % i == 0) {
            cout << "Not Prime" << endl;
            return 0;
        }
        i = i + 1;
    }
    cout << "Prime" << endl;
    return 0;
}
```

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE
 { .\InputAndOutput }
14
Not Prime
```

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

int main()
{
    int n;
    cin >> n;
    int i = 2;
    while(i < n) {
        if(n % i == 0) {
            cout << "Not Prime" << endl;
            return 0;
        }
        i = i + 1;
    }
    cout << "Prime" << endl;
    return 0;
}
```

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE
 { .\InputAndOutput }
11
Prime
```

Patterns

07 January 2022 20:46

Example :

n = 4 (4 rows, 4 columns)

For every row, print 'row' num. of columns or stars.

here n = 3

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

int main() {
    int n;
    cin >> n;
    int i = 0;
    while(i < n) { // i = 0, 1, 2, 3, ..., n-1  which is basically n times
        int j = 0;
        while(j < n) {
            cout << "* ";
            j = j + 1;
        }
        cout << endl; // after printing one row, we need to enter the next row so endl
        i = i + 1;
    }
    return 0;
}
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
.\InputAndOutput }
3
* * *
* * *
* * *
```

Ye code samajh nahi aa raha

Don't worry, initially thoda difficult lag sakta hai BUT the next video has 18 problems solved end to end so keep going

Example : For n = 3,

```
1 1 1
2 2 2
3 3 3
```

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

int main() {
    int n;
    cin >> n;
    int i = 1;
    while(i <= n) {
        int j = 1;
        while(j <= n) {
            cout << i << " ";
            j = j + 1;
        }
        cout << endl;
        i = i + 1;
    }
    return 0;
}
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
4
1 1 1 1
2 2 2 2
3 3 3 3
4 4 4 4
```