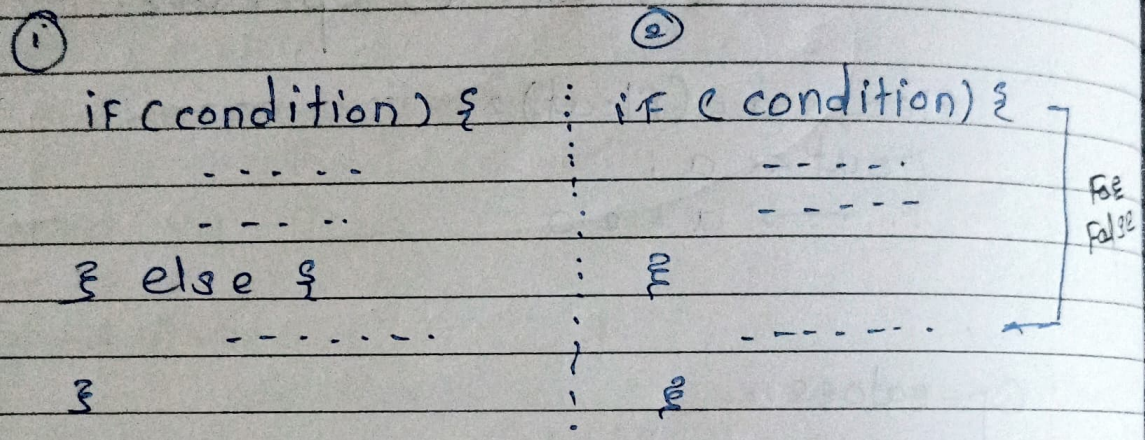
Note:- We don't have to write any condition with the else block.

DAY 3 | 11/5

Conditionals, Loops & Patterns

* Conditionals :-

i) if-else
ii) if-else if-else (Nested if else)



2

* If the condn is true, then execute the line of code within the scope of block.

* If the condn is false, then go for the line just next to the scope of if will be executed

ex:-

```
if (score < 500) {
    cout << "India Wins"
} else {
    cout << "Aust Wins"
}
```

Output :-
true :- India Wins
False :- Aust Wins

```
if (score < 500) {
    cout << "Ind Wins"
}
{
    cout << "Aus Wins"
}
```

Output
true :- Ind Wins Aus Wins
False :- Aus Wins

Note:- We have to write cond^n with if and else if blocks.

∴ Also Note else block is optional if we not write it they will not give any error

ii) Nested if-else

```
if (condition) {
    .....
} else if (condition) {
    .....
} else if (condition) {
    .....
} else {
    .....
}
```

If'g take multiple if else.

Optional.

ex:-

```
if (marks >= 90) {
    cout << "Grade A";
} else if (marks >= 80) {
    cout << "Grade B";
} else if (marks >= 70) {
    cout << "Grade C";
} else if (marks >= 60) {
    cout << "Grade D";
} else {
    cout << "Grade F"
}
```

* Loops:-

Agar hume koi task repeat. repeat karna hai to hum loops use karenge.

- If we want to do same task multiple times then we use loops.

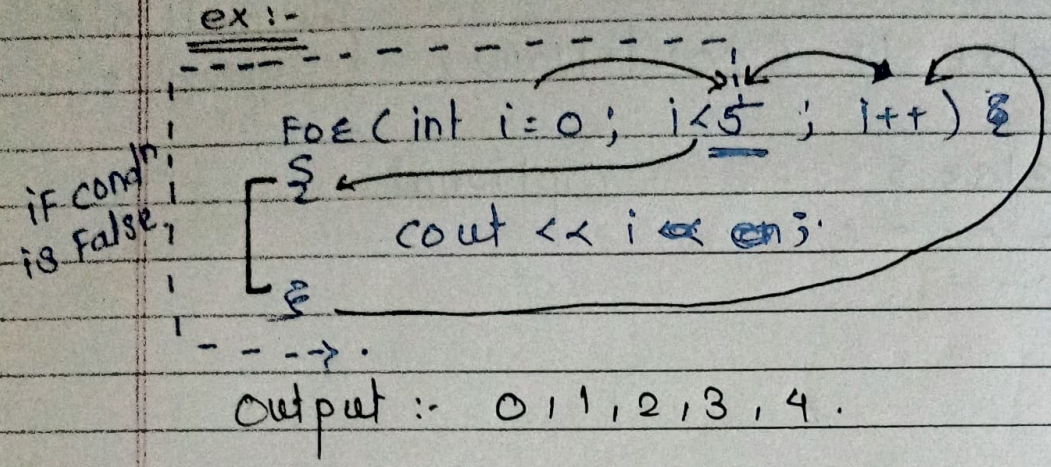
C For loop, while loop, do while loop, For each

Note:- only if is running
only ifelse running
But else will not working
give an error.

initalization Break Condn Updation

```
Foe ( int i=0; i<=10; i++) {
    // Scope of Foe loop.
}
```

ex :-



Output :- 0, 1, 2, 3, 4.

Dry Run :-

(1) i=0
check i<5 true
block { } code Run
Print 0
then go Foe i++ (1)

(2) i=1
check i<5 true
block code { }
print 1
go Foe i++ (2)

(3) i=2
check i<5 true
block code Run { }
Print 2;

go for i++ ③

4. i = 3
check i<5 true
block code Run {}
print 3
go for i++ ④

5. i = 4
check i<5 true
block code Run {}
print 4
go for i++ ⑤

6. i = 5
check i<5 False
then go next line of block code
{ }

ex:-

```
for (int i=5; (i>=0 && i<=10); i++){
        cout<<i;
}
```

only one thing remember multiple cond is used.

output :- 5, 6, 7, 8, 9, 10

∴ Also Another Way of Updation are.

i = i + 2 | i = i/2
i = i * 2 | i = i - 2

Q. Out of initalization, break condition, updation which are optional?

Initalization
Updation
Break condn
} All are optional.

ex:-

```
int i=1;

FOR ( ; ; ){
    if (i<=5){
        cout<<i;
    }   i=i++;
}
```

this will not give any syntax error.

IMP Question What is the output of following code?

i)
```
int n
IF (cin>>n){
        cout << "Bhaiya";
}
```

cout:- For All value of n its print Bhaiya.

ii)
```
int n
if (cout<<n){
    cout << "Bhaiya";
}
```

∴ It's First print the value of n & then print Bhaiya For all +ve, -ve or 0.

Patteens:-

Patteen's are improving logic building of our concept of loops.

How to Approach Patteen Problem

1) First obseeve the no of eows in the patteen.
2) Now obseeve the no of columns in the patteen.
& Building eeln of eow & column.

∴ Note:- Patteens are mostly coded using nested loops.
outee loop:- outee foe loop foe eows
innee loop:- Innee foe loop foe coloumns

Q. Numeeic Full pyeamid

eow=0	_	_	_	_	1				
eow=1	_	_	_	2	3	2			
eow=2	_	_	3	4	5	4	3		foe 'n' value.
eow=3	_	4	5	6	7	6	5	4	
eow=4	5	6	7	8	9	8	7	6	5

i) Space
ii) Fiest Half
iii) second Half

```
#include <bits/stdc++.h>
using namespace std;

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

    for(int row=0; row<n; row++){

        // for space
        for(int col=0; col<n-row-1; col++){
            cout<<" ";
        }

        // Half Pyramid
        for(int col=0; col<row+1; col++){
            cout<<row+col+1;
        }

        // Last Half
        for(int col=0; col<row; col++){
            cout<<2*row-col;
        }
        cout<<endl;
    }
```

Dry Run

row	Coloumn: Space	Half	Last	output
0	0,1,2,3	0		----1
1	0,1,2	0,1	0	---2 3 2
2	0,1	0,1,2	0,1	--3 4 5 4 3

Q.2 Full Pyramid

```
- - - - *
- - - * *
- - * * *
- * * * *
* * * * *
```

i) space
ii) print *

```
For (row=0; row<n; row++) {

space// For( col=0; col< n-row-1; col++) {
            cout << " ";
        }

print // For ( col=0; col<row+1 ; col++){
with          cout << "* " << " ";
space.  }
        cout << endl;
}
```

Q.3 Numeric Half Pyramid

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

```
for(row=0; row<n; row++){

    for(col=0; col<row+1; col++){
        cout << row+1 col+1;
    }                          ↑ column value printed.
    cout<<endl;
```

Q4 Hollow Rectangle.

```
* * * * * * *
*           *
*           *
*           *
* * * * * * *
```

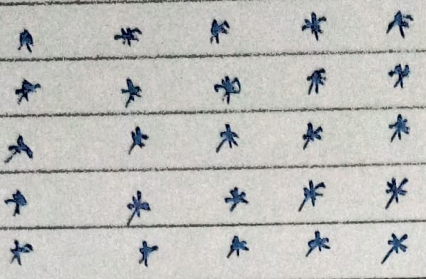
Here row & col take input as m, n.
row=n col=m

```
for( row=0; row<n; row++){

    for(col=0; col<m; col++){

        if( col==0 || col==m-1 ||
            row==0 || row==n-1 ){

            cout<<"*";
        } else { cout<<" " }
    }
    cout<<endl;
}
```

for * print

else space

Q.5 Square Solid



```
for (int row=0; row<n; row++) {
    for (int col=0; col<n; col++) {
        cout << " * ";
    }
    cout << endl;
}
```