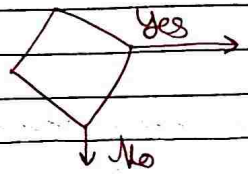
DATE 29/01/23
PAGE NO. :

Third Lecture

Conditions:- This is similar to the decision making box that we studied in flowcharts.




```
if statement
if (condition) {
    ~~~~~~~~~~
}
```

```
if (age > 17)
{
    You are eligible to vote
}
```

If the conditions is true, then execute the line of code within the scope of blocks.

Ex.

```
if (marks > 95)
{
    Cout << "A grade.";
}
```

True → Then execute

Ques: Predict the output of following code:

```
if (score < 300) {
    Cout << "India wins";
}
Cout << "Pak. wins";
```

(i) score = 383 Pak wins

(ii) score = 123 India wins Pak wins

DATE : / /
PAGE NO. :

Note → If Condition is false, then the line just nexet to the scope of if will be executed.

But this looks illegical & hence if else can be used

Exe:-
```
int main() {
    int score;
    Cout << "Enter the score" << endl;
    Cin >> score;
    if(scored < 300) {
        Cout << "India wins" << endl;
    }
    Cout << "Pak wins" << endl;
}
```

The Control will go inside blocks only when Condition is false

(I) Condition → Score = 383 Pak wins
(II) Condition → Score = 183 India wins

What to do if there are multiple Conditions. We will be using if else if else if statements other Solution can be the nested if else

```
if (Condition) { }
else {
    if (Condi2 { }
    else { }
}
```

→ One with in another Nested if else for multiple Conditions.

DATE : / /
PAGE NO. :

Ex :- int age;
Cout << "Enter the age" << endl;
Cin >> age;

if(age >= 18) {
Cout << "I Can Vote" << endl;
}
else {
Cout << "I Can not Vote" << endl;
}

But more readable will be using if else if and else blocks together.

④ if else if else
① if (cond1) { }
② else if (cond 2) { }
③ else { }

Eg:- Grading System.

```
int marks;
Cin >> marks;
if (marks >= 90) {
  Cout << "A grade";
}
else {
  if (marks >= 80) {
  Cout << "B Grade";
  }
else {
  if (marks >= 60) {
  Cout << "C grade";
  }
else {
  if (marks >= 40) {
    Cout << "D grade";
  }
else {
  Cout << "F";
  }
}
}
}
```

DATE : / /
PAGE NO. :

2nd way of Grading System :-

```
if (marks >= 90) {
  cout << "A";
}
else if (marks >= 80) {
  cout << "B";
}
else if (marks >= 60) {
  cout << "C";
}
else if (marks >= 40) {
  cout << "D";
}
else {
  cout << "F";
}
```

But more readable will be using if else if and else blocks together.

```
if else if else
1. if (cond1) {   }
2. else if (cond2) {   }
3. else {   }
4. ...
```

DATE : / /
PAGE NO. :

Flow of above code execute the block 3 :-

- If Condition is true, then go to line 4 but if Condition is false, then check for Condition 2

- If Condition 2 is true, then execute the block 3 go to line 4 but if Condition 2 is false then we will have to execute the else blocks.

⇒ **Note:-** we don't have to write any Condition with the else blocks

- We have to write the Condition with if and else if blocks.

Also we have to remember, else block is optional as if we don't write it then we don't be getting any syntax errors.

⇒ **Note:-**
```
if( ) { 3
else if( ) { 3
else if( ) { 3
else { 3 → optional
```

Also there can be multiple if blocks.

DATE : / /
PAGE NO. :

Ex:- int boonum;
cin >> "boonum;
if(boonum == 0)
{
cout << " Baat ban jayegi ";
}
else
{cout << baat nahi banegi;
}
Op is then we enter 0 = baat ban jayegi
then we enter 1 = baat nahi banegi;

☆ Loops :- We have various type of loops such as ① For loop, ② While loop, ③ Do while loop and ④ Foreach loop but for now we will discuss only for loop.

Loops are used when we want to do same task multiple times.
Ex:- Printing name 5 times, Counting from 1 to 5 etc

For loop Syntax :-

for(int i=0; i<5; i=i+1) {

Scope of for loop

}

DATE : / /
PAGE NO. :

Ex.

① ② False

```
for(int i=0; i<5; i=i+1)
{
  Cout << " Hello Dinesh";
}
```

for (i=0, i<5, i=i+1)

i=0
i=1
i=2
i=3
i=4
i=5 → false

i=0
0<5 → True
1 Hello Dinesh
i=0+1
⇒1
1<5 → True
2 Hello Dinesh
i=1+1=2
2<5 → True
3 Hello Dinesh
i=2+1=3
3<5 → True
4 Hello Dinesh
i=4+1=5
5<5 → False

② for(int i=0; i<3; i=i+1)
{
Cout << i;
}
exit

i=0
0<3 →
1 Print 0
i= i+1=0+1=1
1<3 → True
2 - Print - 1
i= i+1=1+1=2
2<3 → True
3. Print → 2
i=i+1=2+1=3
3<3 - false

DATE : / /
PAGE NO. :

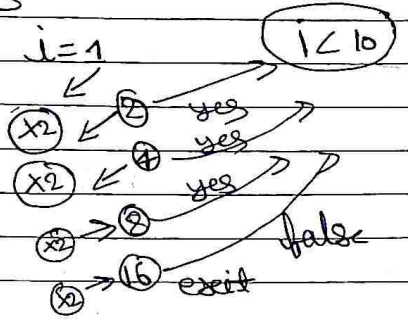
③
```
for(int i=5; i>0; i=i-1){
    cout << i << endl;
}
```

④
```
for(int i=1; i<=10; i=i+1){
    cout << 2*i << endl;
}
```

⑤
```
for(int i=0; i<=5; i=i+2){
    cout << i << endl;
}
```
OP- ~~1 3 5~~
0 2 4

⑥
```
for(int i=1; i<=5; i=i+2){
    cout << i << endl;
}
```
OP 1 3 5

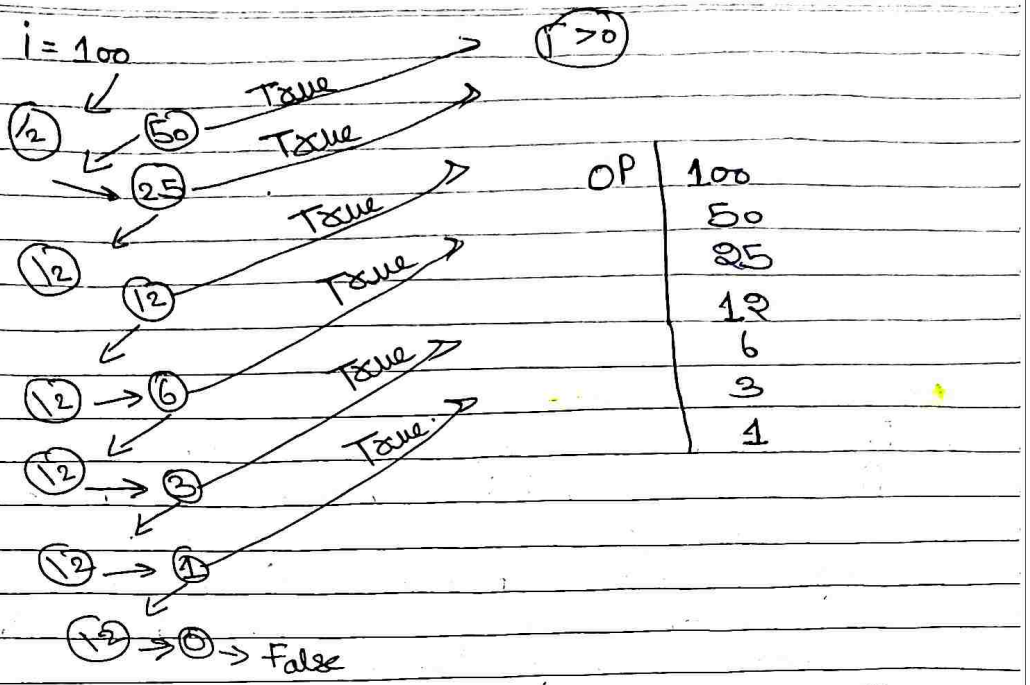
⑦
```
for(int i=1; i<10; i=i*2){
    cout << i << endl;
}
```



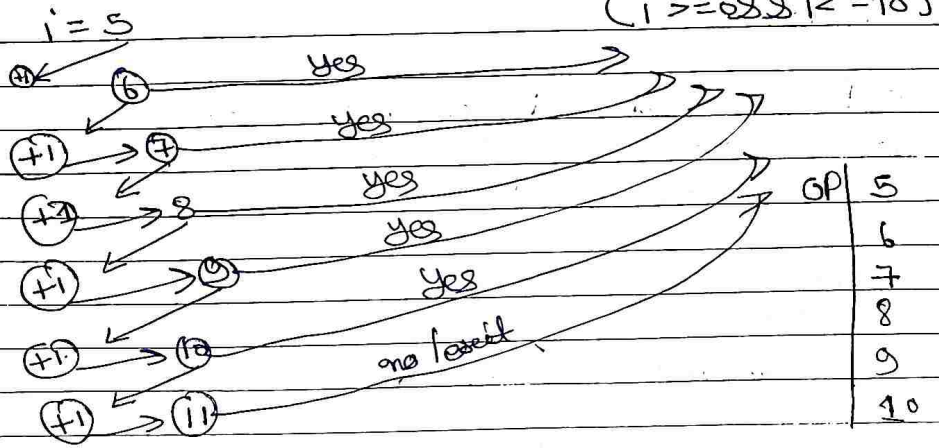
OP 1 2 4 8

⑧
```
for(int i=100; i>0; i=i/2){
    cout << i << endl;
}
```

DATE : / /
PAGE NO. :



② for(int i=5; (i>=0 && i<=10); i=i+1){
Cout << i << endl;
}



DATE : / /
PAGE NO. :

(10)
```
int i=0;
for(;;){
    if(i<5)
    {
    cout<<i<<endl;
    i=i+1;
    }
```
op is = 0, 1, 2, 3, 4

Flow of for loop:-

1) The initialization of variable takes place.
2) The Condition will be checked & if the Condition is true then flow goes inside for loop.
3) After execution, updation & variable takes place & then if the Condition is true, then only flow goes inside for loop.
4) This will go on until the Condition becomes false.

Also note that there will be only one time initialization.

Ques → Predict the o/p of below code

```
for(int i=0; i<3; i++){
    cout<<i;
}
```

output

0 1 2

⇒ Note → updation can be done in any way say

i = i+2 i = i * 2

i = i+1 i = i/2

There are many others ways of updation as well.

DATE : / /
PAGE NO. :

ques → Predict the o/p of below code:

```
for (int i=5; (i)>=0 && i<=10); (i=i+1) {
        cout << i;
    }
```

op is = 5, 6 7 8 9 10

From the above question, we get to know that there can be only conditions.

ques :- out of initialization, updation & condition, which of them are optional & which are mandatory?

Initialization
Condition
updation
} → All are optional

```
for (;;) {
    if (i<5) {
    cout << i;
    i=i+1;
    }
```

→ This will not give any syntax error.

ques → what will be o/p of following code?

```
(1) int n;
    if (cin >> ) {
        cout << "Dinesh";
    }
```

This will always print Dinesh for all values.

DATE : / /
PAGE NO. :

(ii)
```
int n;
cin>>n;
if(cout<<n){
    cout<<"Dinesh";
    3
```

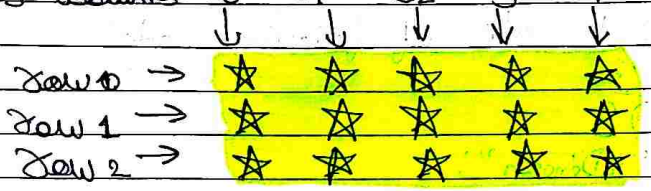
⇒ This will first print the value of n & then, print Dinesh for all value be it -ve, +ve or 0.

Patterns :-

These provide warm up for logic building & will improve your Concept of loops.

Steps to print patterns :-

1) Print observe the number of rows in the pattern. These have 3 rows
2. Now we have to observe the no. of columns & build a relation b/w no. of rows and no. of columns. In this pattern, every column have 3 stars and there are 5 columns. C_0 C_1 C_2 C_3 C_4



Patterns are mostly coded using nested loops. There will be outer & inner for loop. Outer for loop will be for rows & inner loop will be for columns.

DATE : / /
PAGE NO. :

Code :-

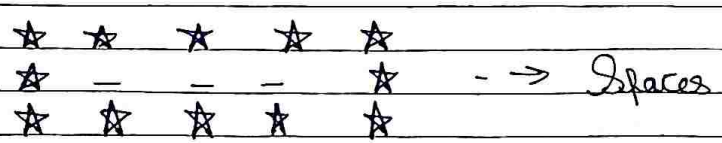
```
for(int row = 0; row < 3; row = row+1){
    // Inner loop
    for(int col = 0; col < 5; col = col+1){
        cout << "*";
    }
    cout << endl; → To go to the next line
```

Note :- row = row + 1 is similar to row += 1

Square pattern :-

In this no. of rows & no. of columns will be same.

Hollow rectangular pattern



1st and last row → 5 stars

other rows → 1st Space last

↓ ↓

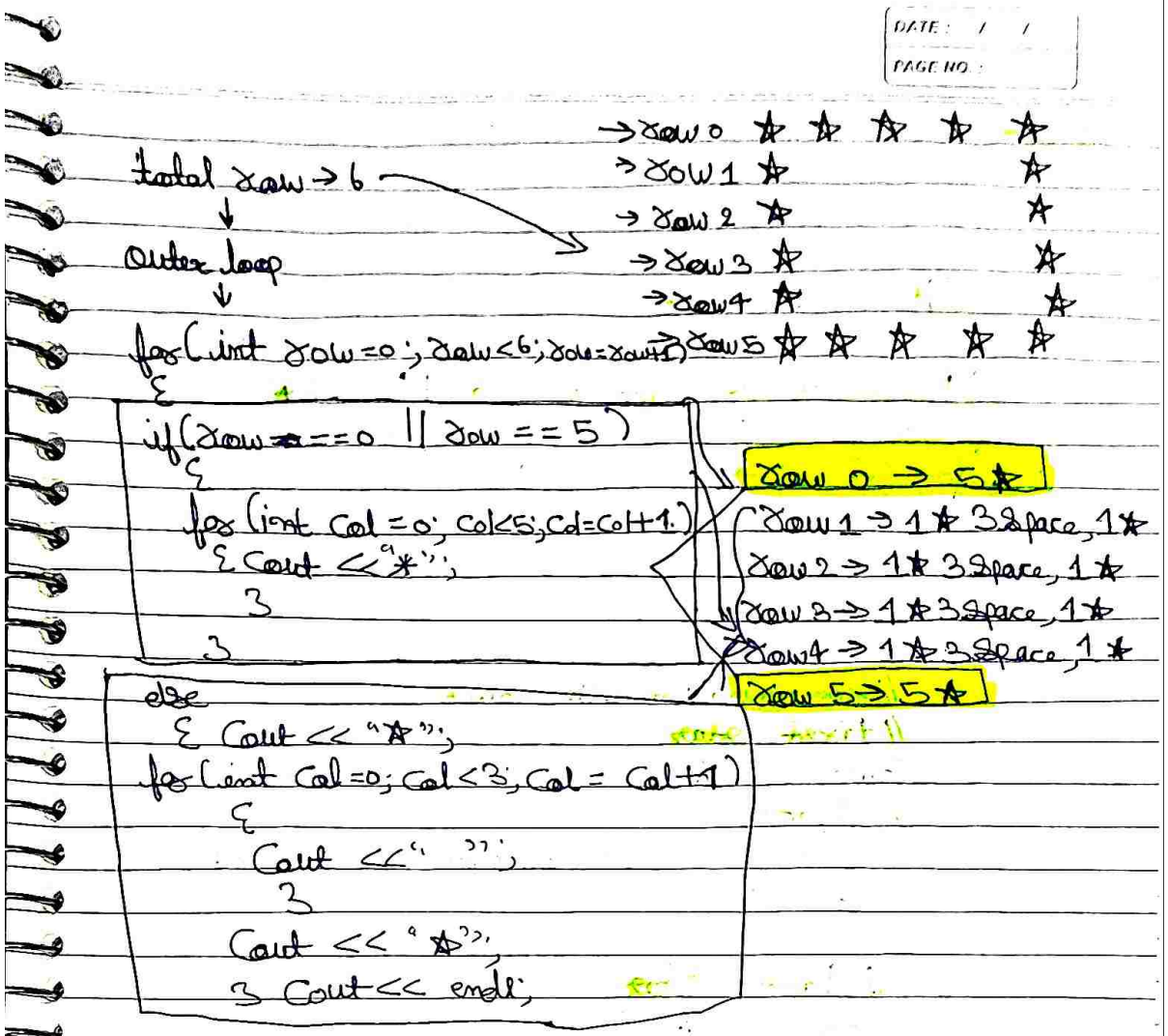
Star Star

)

```
for(int i=0; i<3; i+1){
    for(int j=0; j<5; j=j+1){
        if(i==0 || i==2){
            Cout << "*";
        }
        else{
            if(col==0 || col==4){
                Cout << "*";
            }
            else{
                Cout << " ";
            }
        }
    }
    Cout<<endl;
```

Here i is row
j is column.

⇒
```
int n;
Cin>>n;
for(int row=0; row<n; row+=1){
    for(int col=0; col<n; col+=1){
        Cout << "*";
    }
    Cout <<endl;
}
```


total row → 6
Outer loop
for(int row=0; row<6; row=row+1)
{
if(row==0 || row == 5)
{
for (int col=0; col<5; col=col+1)
{ Cout << "*";
3
3
else
{ Cout << "*";
for(int col=0; col<3; col = col+1)
{
Cout << " ";
3
Cout << "*";
3 Cout << endl;
→ row 0 * * * * *
→ row 1 * *
→ row 2 * *
→ row 3 * *
→ row 4 * *
→ row 5 * * * * *
row 0 → 5*
row 1 → 1* 3 space, 1*
row 2 → 1* 3 space, 1*
row 3 → 1* 3 space, 1*
row 4 → 1* 3 space, 1*
row 5 → 5*

DATE : / /
PAGE NO. :

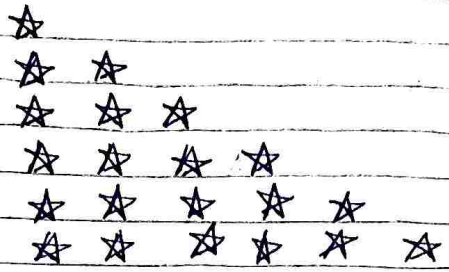
Code :-

```
int rowCount, colCount;
Cin >> rowCount;
Cin >> colCount;

// Hollow Rectangle
for(int row=0; row<rowCount; row=row+1){
// first row or last row -> Print 5*
if(row ==0 || row == rowCount-1) {
for(int col=0; col< colCount; col++) {
cout << "*";
}
}
else {
// remaining middle rows
// first star
Cout << "*";
// spaces
for(int i=0; i< colCount-2; i=i+1) {
Cout << " ";
}
// last star
Cout << "*";
}
Cout << endl;
}
```

DATE : / /
PAGE NO :

Half Pyramid :-

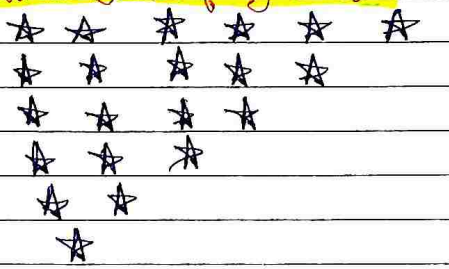


⇒ Number of stars to print is one more than row number.
Also no. of stars are equal to the no. of colums in that particular row.

Code :-

```
// number of rows
int n;
Cin >> n;
for (int row = 0; row < n; row = row + 1) {
for (int col = 0; col < row + 1; col = col + 1) {
    Cout << endl; << "*";
}
} Cout << endl;
}
```

Inverted half Pyramid :-



→ row → 0
→ row → 1
→ row → 2
→ row → 3
→ row → 4
→ row → 5

DATE : / /
PAGE NO. :

row 0 → 6 * → 6-0
row 1 → 5 * → 6-1
row 2 → 4 * → 6-2
row 3 → 3 * → 6-3
row 4 → 2 * → 6-4
row 5 → 1 * → 6-5

⇒ In the code of previous code we just have to change condition in inner for loop
Col < n - row → No. of rows in this question.

Code:

```
// number of rows
int n;
Cin >> n;
for(int row=0; row<n; row=row+1){
for(int col=0; col<n-row; col=c+1){
Cout << "*";
}
Cout << endl;
}
```

3
3
3

Numeric half Pyramid

```
1 → r-0          n=5
1 2 - r-1
1 2 3 - r-2
1 2 3 4 - r-3
1 2 3 4 5 → r-4
```

```
for(int row=0;
    row<n;
    row=row+1)
for(int col=0;
    col<row+1
    col=c+1
{ Cout << endl;
```

Observation :-
row 0 → 1
row 1 - 12
row 2 → 123
row 3 - 1234
row 4 - 12345
Col1 Col2 Col3 Col4 Col5

DATE : / /
PAGE NO. :

Code :

```
int n;
Cin >> n;
for(int row=0; row<n; row=row+1){
    for(int col=0; col<row+1; col=col+1){
        Cout << col+1;
    }
    Cout << endl;
}
```

3
3

In the inverted half pyramid pattern just change
Cout << "*" to
Cout << col+1;