Lecture - 3

Conditional :- It is used as a condition where we use decision block in flowchart

Conditional statement are :-

if - else statement

Ex :- if (condition = true)

```
{
    body execute
}
```

```
if (marks > 95)
{ print A grade;
}
```

```
if (condition = true)
{
}
else (condition != true)
{
}
```

Q-1- Make program of student Grade?

Ans/-

Done in Lec 1

Code in Lec 1 code file

Method to write if else

① if {
}

② if {
}
else
{ if {
}
else {
}
}

③ if {
}
else if {
}

④ if {
}
else if {
}
else if {
}

Q# No. of bus

```
int main(){
 cout << "Enter the brothers" << endl;
 int busnum;
 cin >> busnum;
 if (busnum == 0){
  cout << "Baat Bangi" << endl; }
 else {
 cout << "Baat nhi Bani" << endl;
 }
```

Loops :- Loops are used when we want execute same lines of code again & again

There are 4 types of loops.

1. for
2. while
3. do-while
4. for each

Syntax of for loop :-

for (——— ; ——— ; ———)

↓ initialize ↓ condition ↓ increment / decrement

Ex:- for (int i=0 ; i<5 ; i=i+1)

```
{
    cout << "Harsh" ;
}
```

Q- WAP to make a table of n number?

Ans :-

Done in Lec3 code

→ int i=5

if (i>=0 && i<=10) o/p - 5

i will run for 5, ●, ●●●●●●●●

→ Ex :- for (int i=0 ; i<3 ; i=i+1)

```
{ cout << i;
}
```

o/p - 0 1 2

for (int i=5; i>0; i=i-1) o/p 5 4 3 2 1
{ cout << i << endl;
}

for (int i=2; i<=10; i=i+1) {
cout << 2*i << endl;
}
o/p - 2, 4, 6, 8, 10, 12, 14, 16, 18, 20

for (int i=0; i<=5; i=i+2)
{ cout << i << endl;
}
o/p - 0, 2, 4

for (int i=2; i<=5; i=i+2) {
cout << i << endl;
}
o/p - 1, 3, 5

for (int i=1; i<10; i=i*2) {
cout << i << endl;
}
o/p - 1, 2, 4, 8

for (int i=100; i>0; i=i/2) {
cout << i << endl;
}
o/p - 100, 50, 25, 12, 6, 3, 1

for (int i=0; (i>=5 && i<=10); i=i+1) {
cout << i << endl;
}
o/p - Blank

Q What are mandatory in four loop?

Ans No nothing is mandatory in four loop

four (; ;)

intialization & condition & updation intialie in body.

Q- What its output?

(a) int n;

if (cin >> n)

{ cout << "Haush";

}

n = 2, -4

Ans:- Haush

It print only one time.

Q-(b) int n;

if (cout << "Babbar")

{ cout << "Love";

}

Ans:- Babbar Love

It print both statements if condition become true. That's why it print both parts.

→ Row ↓ column

Patterns :- Patterns increase our logical building.

① → Solid Rectangle :-

R0 → * * * * *
R1 → * * * * *
R2 → * * * * *
col0 col1 col2 col3 col4

total row → 3
col 0 → 3 *
col 1 → 3 *
col 2 → 3 *
col 3 → 3 *
col 4 → 3 *

→ Always use
① outer loop (rows)
② Inner loop (columns)

```
for (int row=0; row<3; row++){
  for (int col=0; col<5; col++){
    cout << "*";
  }
  cout << endl;
}
```

② Square Pattern :-

R0 → * * * *
R1 → * * * *
R2 → * * * *
R3 → * * * *
col0 col1 col2 col3

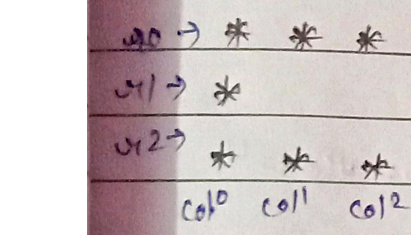
total row → 4
col 0 → 4 *
col 1 → 4 *
col 2 → 4 *
col 3 → 4 *

```
for (int row=0; row<4; row++){
    for (int col=0; col<4; col++){
        cout << "*";
    }
    cout << endl;
}
```

→ Points to Remember:-

① row & column count (first row then column)
② row break
③ next line
* print

③ Hollow Rectangle:-



row 0 → 5*
row 1 → 1*, 5*, 2,3,4 space
row 2 → 5*

```
for (int row=0; row<3; row++){
  if (row==0 || row==2){
      cout << "*";
    for(int col=0; col<5; col++){
        cout << "*";
    }
  else{
      cout << "*";          → first star
    for (int i=0; i<3; i++){          → 2 3 4 (0, 1, 2)
        cout << " ";
    }
    cout << "*";          → last star
  }
  cout << endl;
}
```

→ DRY RUN ALL Pattern's code

④ Half Pyramid:-

row 0 → *
row 1 → * *
row 2 → * * *
row 3 → * * * *
row 4 → * * * * *
col 0 col 1 col 2 col 3 col 4

row 0 → 1*
row 1 → 2*
row 2 → 3*
row 3 → 4*
row 4 → 5*
"row++"

```
for (int row=0; row<n; row++){
    for (int col=0; col<row+1; col++){
        cout << "* ";
    }
    cout << endl;
}
```

⑤ Inverted Half Pyramid:- n=5

row 0 → * * * * *
row 1 → * * * *
row 2 → * * *
row 3 → * *
row 4 → *

row 0 → 5*
row 1 → 4*
row 2 → 3*
row 3 → 2*
row 4 → 1*
"n-row"

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

⑥ Numeric Half Pyramid

n=5

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

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

row 0 → 1 col = row+1 }
row 1 → 1 2
row 2 → 1 2 3
row 3 → 1 2 3 4
row 4 → 1 2 3 4 5

⑦ Inverted Numeric Half Pyramid :-

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

row 0 = 1, 2, 3, 4, 5 n=5
row 1 = 1, 2, 3, 4 "n-row"
row 2 = 1, 2, 3
row 3 → 1, 2
row 4 → 1

```
for (int row=0; row<n; row++){
    for (int col=0; col<(n-row); col++){
        cout << col+1;
    }
    cout << endl;
}
```

⑧ Full Pyramid :-

```
- - - - *          n=5   row 0 → 4 space, 1 star
- - - * * *               row 1 → 3 space, 2 star
- - * * * * *             row 2 → 2 space, 3 star
- * * * * * * *           row 3 → 1 space, 4 star
* * * * * * * * *         row 4 → 0 space, 5 star
```

① Space ⇒ n - row - 1

② Star : for (int row = 0 ; row < n ; row ++) { (n - (row + 1))

// space's for (int space = 0 ; space < ; space++

{ cout << " " ;

}

for (int col = 0 ; col < (row + 1) ; col++)

{

cout << "* " ;

space

}

cout << endl ;

}

⑨ Inverted full Pyramid

```
* * * * *      n=5
- * * * *      row 0 → 0 space, 5 star
- - * * *      row 1 → 1 space, 4 star
- - - * *      row 2 → 2 space, 3 star
- - - - *      row 3 → 3 space, 2 star
               row 4 → 4 space, 1 star
```

space → row

star → n - row

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

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

{ cout << " " ;

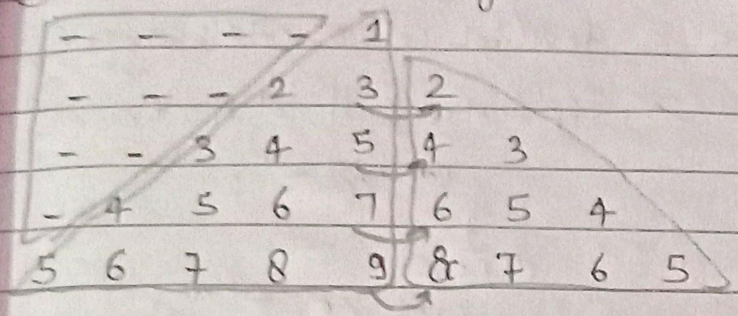
}

for (int col = 0 ; col < n - row ; col++)

{ cout << "* " ;

} }

(10) Numeric full pyramid :-




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

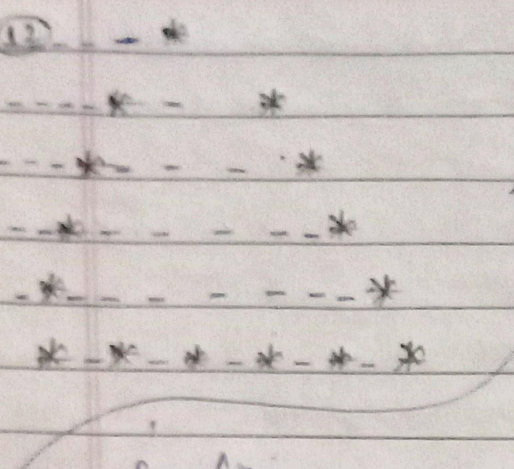
Done (Full Backside)

(11) Numeric Hollow full pyramid :-

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

Done on (Backpage)

(12)



```
        else {
            for(int col=0; col<(2*row+1); col++)
            if(col==0 || col==2*row)
                cout<<"*";
            }
            else{
                cout<<" ";
            }}cout<<endl;}
    for(int row=0; row<n; row++){
        for(int space=0; space<n-row-1; space++)
            cout<<" ";
        }
        if(row==n-1){
            for(int star=0; star<(2*row+1); star++)
                if(star%2==0){
                    cout<<"*";}
                else{cout<<" ";}}}}
```

Do Home work Weekly →
Weekly C++ Quiz →
Debugging / Exercise →

Weekly Revision

Q. What is ternary operator?

Ans:

A ternary operator evaluates the test condition & execute a block of code based on the result of the condition.

Syntax:

condition ? expression 1 : expression 2

what an ouptupt?

① char char = '2'; → 2

② char endl = 'a'; → aa

③ char main = 'c'; → c

④ char int = 'd'; → Invalid combination two identifiers

⑤
```
short a = 2^16 - 1;
short b = 10;
short c = a+b;
```
→ Not output

⑥
```
int a = 5;
char b = 'd';
int sum1 = a+b;
cout << sum1
```
→ 105

```
float f = 2.0 + sum1;
cout << f
```
→ 107

⑦
```
float f = 2.7;
int n = 157;
int diff = n - f;
cout << diff
```
→ 154

→ Lecture - 4 →

13) Solid Diamond