

Lecture 3

Conditionals Loops & Patterns

Date _____
Page _____

Conditionals

* if statement -

We use if statement if we have a condition which can either be true or false. If the condition provided with if statement then the code belonging to it will be executed.

Syntax -

```
if (condition) {  
    }  
}
```

* if else statement -

When if we have a condition for which there are two outcomes then ; if the condition is true ; then code belonging to if statement is executed otherwise code belonging to else statement is executed.

Syntax -

```
if (condition) {  
    }  
}
```

```
else {  
    }  
}
```

If- ~~or~~ else if - else -

It is used when there are multiple conditions.
syntax -

if (condition 1){

 } else if (condition 2){

 } else {

Example → if (marks >= 90){
 cout << "A grade";
 }
 else if (marks >= 80){
 cout << "B grade";
 }
 else if (marks >= 60){
 cout << "C grade";
 }
 else {
 cout << "D grade";
 }

* Loops

- for
- while
- do while
- for each

repeat.

* for loop -

syntax → `for(int i=0; i<5; i++)`

initialize ↓ update
condition

Example - `for(int i=0; i<5; i++) { }`

cout << "Lone Babbar";

3.

Lone Babbar will print
5 times

Processing -

1. $i=0$.

4. $i=3$

$3 < 5 \rightarrow \text{True}$

Lone Babbar

$i=i+1(i++) = 4$

5. $i=4$

$4 < 5 \rightarrow \text{True}$

Lone Babbar

$i=i+1(i++) = 5$

6. $i=5$

$5 < 5 \rightarrow \text{False}$

Loop break

$0 < 5 \rightarrow \text{True}$

print Lone Babbar - ①

$i=i+1(i++) = 1$

2. $i=1$

$1 < 5 \rightarrow \text{True}$

Lone Babbar - ②

$i=i+1(i++) = 2$

3. $i=2$

$2 < 5 \rightarrow \text{True}$

Lone Babbar - ③

$i=i+1(i++) = 3$

Example \rightarrow for (int $i=0$; $i < 3$; $i = i + 1$)

3
cout << i << endl;

Processing -

1. $i = 0$

$0 < 3 \rightarrow$ True

print 0

$i = 0 + 1$

2. $i = 1$

$1 < 3 \rightarrow$ True

print 1

$i = 1 + 1$

3. $i = 2$

$2 < 3 \rightarrow$ True

print 2

$i = 2 + 1$

4. $i = 3$

$3 < 3 \rightarrow$ False

loop break

Output -

0
1
2

Example \rightarrow for (int $i=0$; $i < 5$; $i = i + 1$).

3
cout << i << endl;

Processing -

1. $i = 0$

$0 < 5 \rightarrow$ True

print 0

$i = 0 + 2 = 2$

2. $i = 2$

$2 < 5 \rightarrow$ True

print 2

$i = 2 + 2 = 4$

3. $i = 4$

$4 < 5 \rightarrow$ True

print 4

$i = 4 + 2 = 6$

4. $i = 6$

$6 < 5 \rightarrow$ False

loop break

O/P - 0
2
4

Date _____
Page _____

Example → `for(int i=0; i<10; i=i*2){
 cout << i << endl;
}`

Processing =

1. $i = 0$
 $0 < 10 \rightarrow T$
print 1
 $i = 0 * 2 = 2$

2. $i = 2$
 $2 < 10 \rightarrow T$
print 2
 $i = 2 * 2 = 4$

3. $i = 4$
 $4 < 10 \rightarrow T$
print 4
 $i = 4 * 2 = 8$

4. $i = 8$.
 $8 < 10 \rightarrow T$
print 8
 $i = 8 * 2 = 16$

5. $i = 16$
 $16 < 10 \rightarrow F$
Loop break

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

Processing -

1. $i = 100$
 $100 > 0 \rightarrow T$

$$i = 100/2 = 50$$

2. $i = 50$
 $50 > 0 \rightarrow T$

$$i = 50/2 = 25$$

3. $i = 25$
 $25 > 0 \rightarrow T$

$$i = 25/2 =$$

Date _____
Page _____

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

3 cout << i << endl;

Processing -

1. i = 5

$5 \geq 0 \& 5 \leq 10 \rightarrow T$

print 5

$i = 5 + 1 = 6$

2. i = 6

$6 \geq 0 \& 6 \leq 10 \rightarrow T$

print 6

$i = 6 + 1 = 7$

3. i = 7

$7 \geq 0 \& 7 \leq 10 \rightarrow T$

print 7

$i = 7 + 1 = 8$

4. i = 8

$8 \geq 0 \& 8 \leq 10 \rightarrow T$

print 8

$i = 8 + 1 = 9$

5. i = 9

$9 \geq 0 \& 9 \leq 10 \rightarrow T$

print 9

$i = 9 + 1 = 10$

6. i = 10

$10 \geq 0 \& 10 \leq 10 \rightarrow T$

print 10

$i = 10 + 1 = 11$

6. i = 11

$11 \geq 0 \& 11 \leq 10 \rightarrow F$

Loop break

NOTE : Three conditions of your loop
- initialize, condition and updations
all are optional.

Homework

①

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

In this question Hi will be printed only once.

Let's say, we take 4 as input in n, so-

if ($\text{cin} \gg 4$) \rightarrow this condition is true
so it will print
Hi once

whatever number we take in it whether it is positive, negative or zero, if condition can't be false, it will be always true and hence it will print Hi once

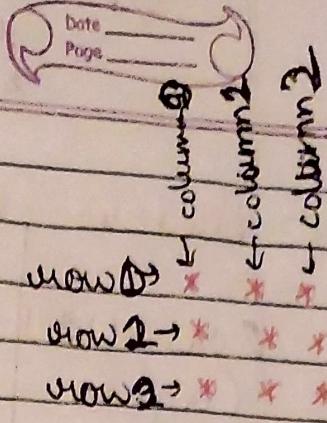
②

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

In this question output will be OHi because the first cout is $\text{cout} << n$, so we don't have a value for n so it will print 0

and second cout will print Hi because condition will always be true

Patterns



1. Row observation.

- total no. of rows

row \Rightarrow

row 1 \rightarrow

row 2 \rightarrow

row 3 \rightarrow

2. Column observation

- total no. of columns

① Solid Rectangle

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

There are three rows, so,

```
for (int row=0; row<3; row=row+1)
{
    for (int col=0; col<5; col=col+1)
        cout << "*";
}
```

Now, there are five columns so,

```
for (int row=0; row<3; row=row+1)
{
    for (int col=0; col<5; col=col+1)
        cout << "*";
}
```

O/P -



Date _____

Page _____

Day run -

2. $row = 1, 1 < 3 \rightarrow T$

(again
these
same
steps
will follow
of
inner loop
which was
followed
previously)

$$row = 1 + 1 = 2$$

and
similarly
so on

till the
loop breaks.

1. $row = 0, 0 < 3 \rightarrow T$
 $(col = 0, 0 < 5 \rightarrow T)$



$$col = 0 + 1 = 1$$

$$col = 1, 1 < 5 \rightarrow T$$



$$col = 1 + 1 = 2$$

$$col = 2, 2 < 5 \rightarrow T$$

$$col = 2 + 1 = 3$$

$$col = 3, 3 < 5 \rightarrow T$$

$$col = 3 + 1 = 4$$

$$col = 4, 4 < 5 \rightarrow T$$



$$col = 4 + 1 = 5$$

$$col = 5, 5 < 5 \rightarrow F$$

Loop break

$$row = 0 + 1 = 1$$

→ Inner
loop

②

Square pattern -

There are 4 rows
so,

* * * * ← row0
* * * * ← row1
* * * * ← row2
* * * * ← row3

```
for (int row=0; row<4; row=row+1)
```

```
    cout << "*";
```

{

Now, there are 4 columns, so,

```
for (int row=0; row<4; row=row+1)
```

```
    for (int col=0; col<4; col=col+1)
```

```
        cout << "*";
```

{

}

* Hollow Rectangle



// There are 4 rows.

```
for(int row=0; row<4; row=row+1)  
{
```

// We want 5 stars in first and
last row.

```
if (row==0 || row==3);
```

{ // There are 5 columns.

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

cout << "*";

}

y

else

{

cout << "*";

```
for(int col=1; col<2; col=col+1)  
{
```

cout << " ";

}

cout << "*";

y

}

* Half Pyramid -

★	→ row 0 = 1 ★
★★	→ row 1 = 2 ★
★★★	→ row 2 = 3 ★
★★★★	→ row 3 = 4 ★
★★★★★	→ row 4 = 5 ★
★★★★★★	→ row 5 = 6 ★

// There are 6 rows.

```
for (int row=0; row<6; row=row+1)
```

// There are n st!

/* The number of stars on the
number of row is same

as 1st row or row 0
containing 1★, 2nd row
contains 2★ and so on */

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

 cout << "★";

3. cout << endl;

* Inverted Half Pyramid -



→ row 0 = 6 *

→ row 1 = 5 *

→ row 2 = 4 *

→ row 3 = 3 *

→ row 4 = 2 *

→ row 5 = 1 *

// There are 6 rows.

```
for (int row = 0; row < 6; row = row + 1)
```

* It is following a formula, i.e.,
total no. of stars - selected row, so */
for (int col = 0; col < 6 - row; col = col + 1)

```
cout << "*";
```

y

```
cout << endl;
```

g

* Numeric Half Pyramid -

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

// There are 5 rows.

```
for(int row=0; row<5; row=row+1)
```

{
* col is following a formula
of no. of spaces
row no. + 1 */

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

```
cout << col+1;
```

}

```
cout << endl;
```

}

* Numeric Inverted half pyramid-

```

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

```

// There are 5 rows

for (int row=0; row<5; row=row+1)

for (int col=0; col<5-row; col=col+1)

cout << col+1;

}

} cout << endl;

* Full Pyramid-

```

*
 *
 *
 *
 *
 *
 *
 *
 *
 *
 *
 *
 *
 *
 *
 *
 *

```

Homework