URBAN EDGE

Chapter 4 - Conditionals in Java

Sometimes we want to watch comedy videos on YouTube if the day is Sunday.
Sometimes, we order junk food if it is our friend's birthday in the hostel
You might want to buy an Umbrella if its raining and you have the money.
You order the meal if aloo or your favorite bhindi is listed on the menu.

All these are decisions which depends on a certain condition being met.
In Java, we can execute instructions on a condition being met.

Decision making Instructions in Java
→ If - Else Statement
→ Switch statement

If - else Statement
The syntax of an If - Else statement in C looks like that of C++ and JavaScript. Java has a similar syntax too. It looks like:

```
if (condition-to-be-checked) {
        Statements-if-Condition-true;
    }
else {
        Statements-if-Condition-false;
    }
```

URBAN EDGE

Code Example :

```
int a = 29;
if ( a > 18) {
    System.out.println(" You can drive");
}
```

Note that the else block is optional

Relational Operators in Java
Relational operators are used to evaluate conditions (true or false) inside the if statements.
Some examples of relational operators are :

==, >=, >, <, <=, !=

== → equals
>= → greater than or eq. to
!= → Not equals

Note : '=' is used for assignment where as '==' is used for equality check.

The condition can be either true or false.

Logical Operators
&&, || and ! are most commonly used logical operators in Java
These are read as :

&& → AND
|| → OR
! → NOT

⇒ Used to provide logic to our JAVA programs

URBAN EDGE

AND operator

Evaluates to true if both the conditions are true

```
Y && Y = Y
Y && N = N
N && Y = N
N && N = N
```

Y → true
N → false

OR Operator

Evaluates to true when at least one of the conditions is true.

```
Y || Y = Y
Y || N = Y
N || Y = Y
N || N = N
```

Y → true
N → false

NOT Operator

Negates the given logic (true becomes false and false becomes true)

```
! Y = N
! N = Y
```

Y → true
N → false

else if clause

Instead of using multiple if statements, we can also use else if along with if thus forming an if-else-if-else ladder

Using such kind of logic reduces indents. last else is executed only if all the conditions fail.

URBAN EDGE

```
if (condition1) {
        // Statements;
    }
else if (condition
        // Statements;
    }
else {
        // Statements;
    }
```

Switch Case Control Instruction

Switch - Case is used when we have to make a choice between number of alternatives for a given variable

```
Switch (var) {
    Case C1 :
        // Code;
        break;
    Case C2 :
        // Code
        break;

    Case C3 :
        // Code
        break
    default :
        // Code

}
```

URBAN EDGE

Var can be an integer, character or string in Java.

A switch can occur within another but in practice this is rarely done