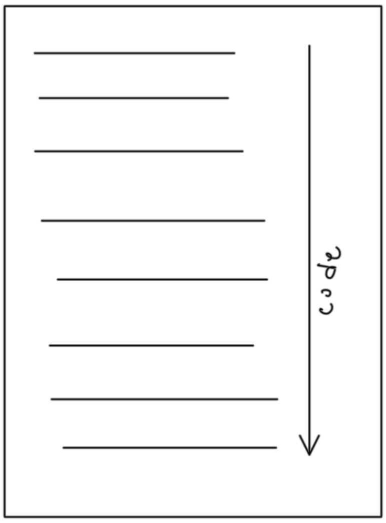
conditional statements

Day 16
26/07/2025

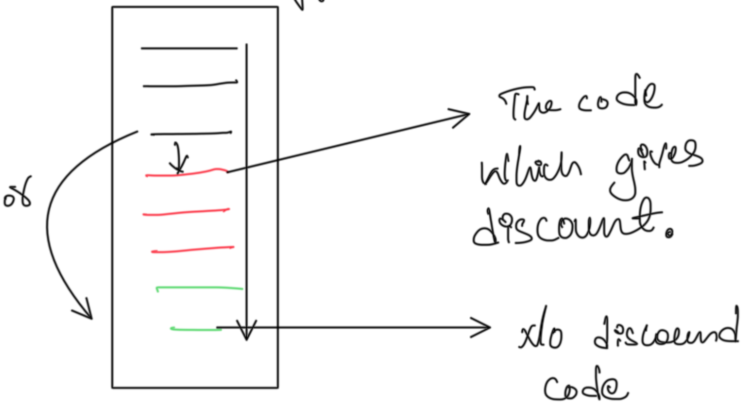
- Why we Need conditional statements?
  logical Bugs, Error handling
- Where its is used in the industry or coding
- Types of patterns of conditional statements
- How to write clear & clean conditional statements
  - if
  - if else
  - if else if else
  - Nested if else
  - Conditional
  - Switch

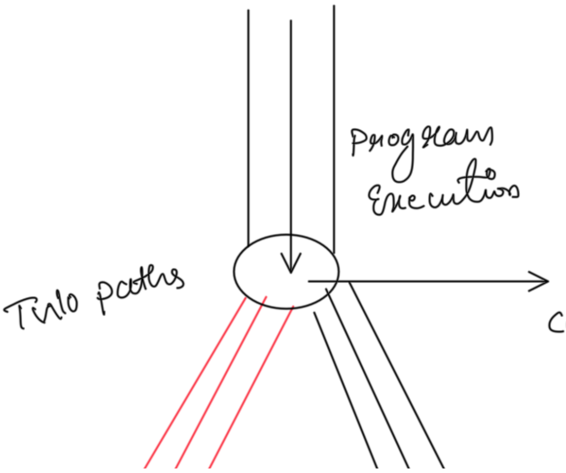
---

Why we Need conditional statements?



Let Imagine a situation where an Ecart online shopping Company wants to give 20% of discount for the customers who are doing shopping 5000 or above.
How to differentiate this?





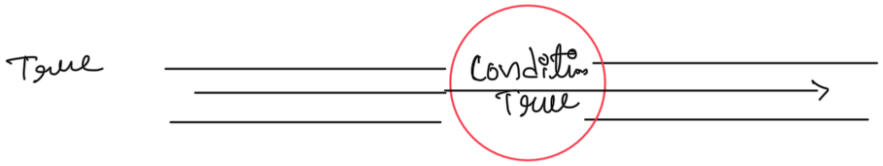
condition (choose any on path)

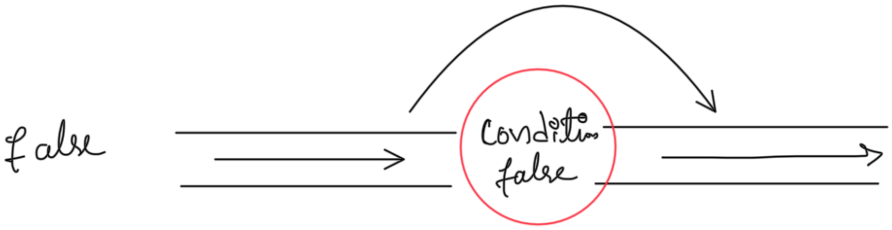
Creating a condition for a task or block of code to execution

block of code executed
Based on condition the CPU performs the operation.

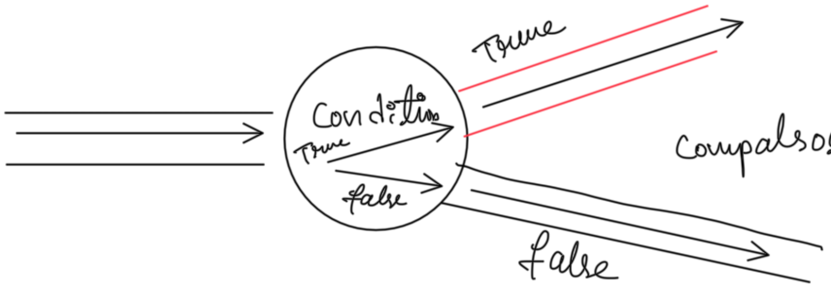
There are few patterns for creating a conditions

# pattern 1 ( if pattern )



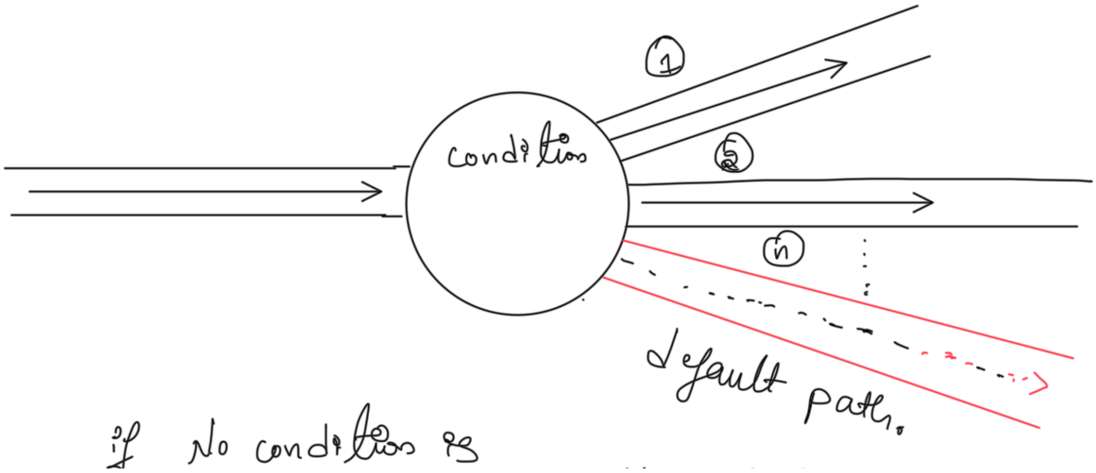


# pattern 2 ( if else pattern )



compulsory have to choose any one path

# pattern 3 ( More than two paths )



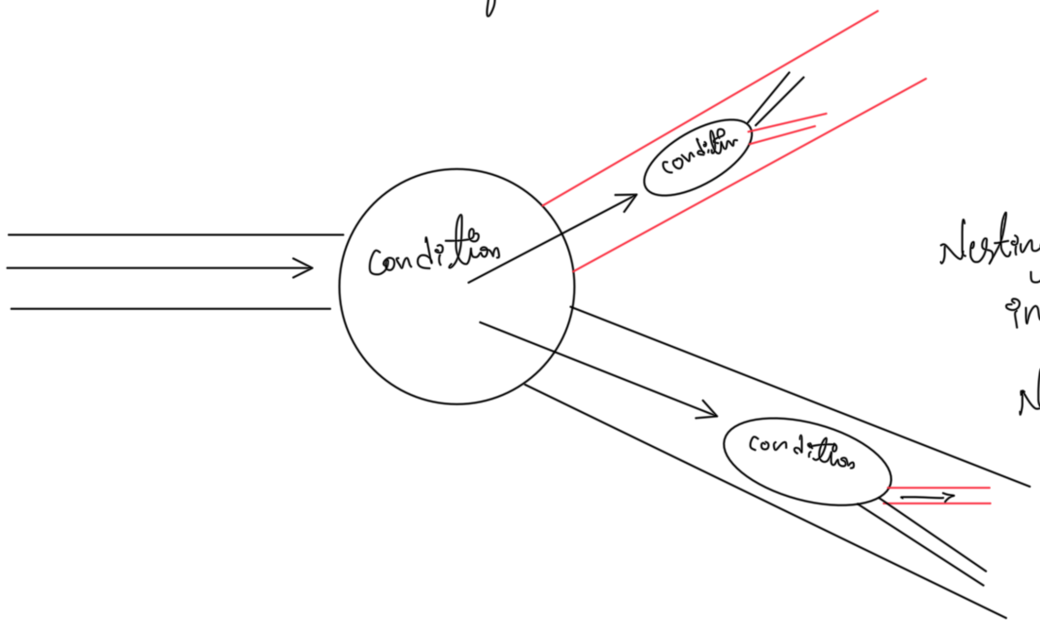
chooses any one path amo

if No condition is

true then the default path is choosen

# pattern 4

Nesting of pattern 2 (Nested if else)



Nesting of patterns inside pattern 2

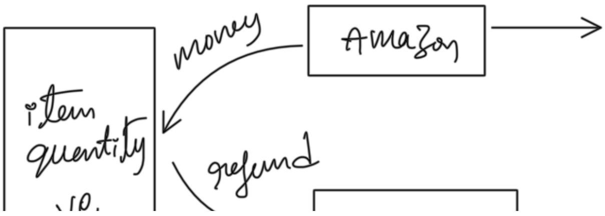
Nesting if else

Most of the Cases the programmers do mistakes in the conditional statements that Causes some issues in Program Execution or Result.

Example

Shopping Cart Bug Causes Millions of Dollors loss to the Amazon Company.

The user selects items as -ve quantity unlike 1, 2, 3 the amount will be refunded to the user from amozon Bank account.

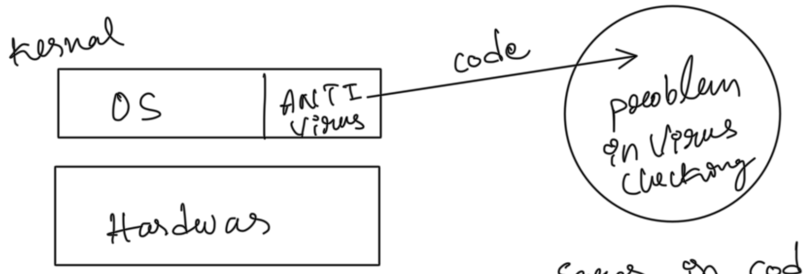


Millions of Dollors loss for amazon Company.

amazon cart

(2) Blue screen problem in windows

Crowd strike Antivirus Company.



Errors in code Causes windows operating System crash.

It treated as a world worst down time

which Causes Millions of dollers of loss

So Handling Each possible condition in the program while Execution plays an important role.

---

Conditional statement

(condition) ? $\boxed{\text{True}}$ : $\boxed{\text{False}}$ ;

similar to If else statement

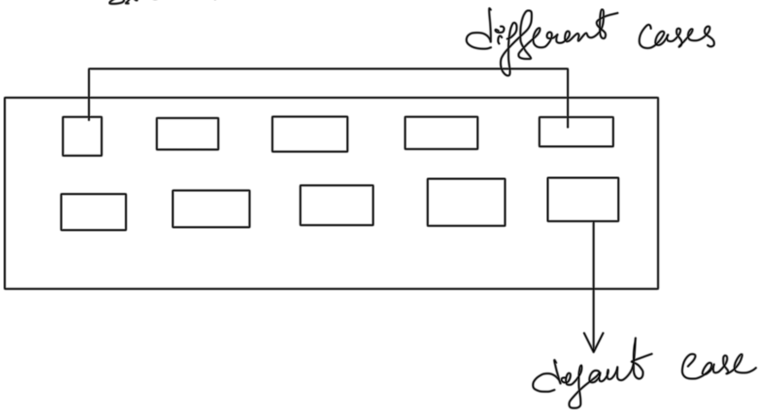
```
if (num > 0) {
    print ("positive")
}
else {
    print ("Negative")
```

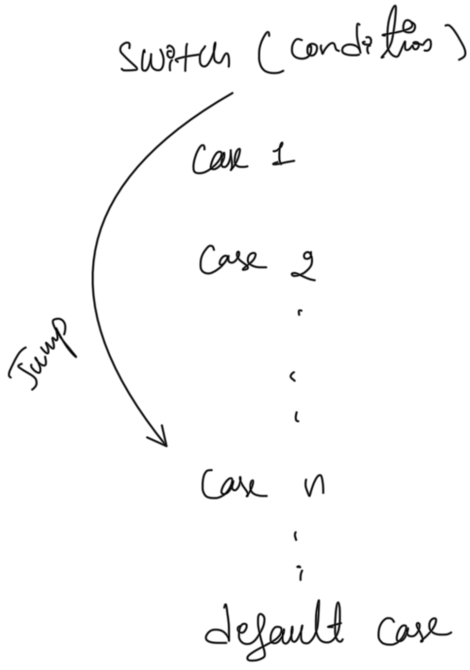
```
Print ( (num >0) ? "positive : "Negative");
```

short hand of if else condition

}

Switch statement




switch (conditions)

case 1

case 2

⋮

case n

⋮

default case

Instead of checking Each and Every Conditions, it directly Jump to the condition.

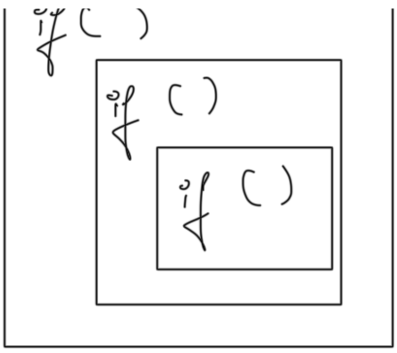
case statement cant Evaluate Expressions as the case.

default case is Executed if no case found or matches

How to Write class & Clean conditions

① Avoid Nesting of conditions

Nesting of conditions makes debugging Hard and decreases the readability of code.



② use Early Exit's by checking the inputs

to avoid "if else nested code Execution"

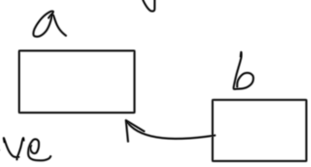
③ avoid Writting More Expressions to Evaluate as a condition

Example  if( cond1 && cond2 || cond3 && && cn3 || con4 || con5 && &&)

④ == & = Bug in c program compilation

if(a==b) ✓  if(a=b)  assigns b to a & if a is +ve then if Block will be Executed.

```
if(a=b) {
   ≡   ] Execute
           if a is +ve
}
```



# Understanding Conditional Statements in Java

## 1. What Are Conditional Statements?

Conditional statements control the flow of a program by executing different code depending on whether a condition is true or false.

## 2. Why Do We Need Conditional Statements?

Without conditional logic, a program would run in a fixed, linear way. Conditionals allow us to model real-world decision-making.

**Examples:**

- Grant login access only if credentials match.
- Offer discount if total amount exceeds a threshold.
- Alert system if temperature crosses safe limits.

## 3. Industrial Use Cases

- **E-commerce (Amazon):** Apply discounts based on conditions like total amount, items, or membership.
- **Banking:** If balance ¡ threshold, impose penalty.
- **Healthcare:** Diagnose based on symptoms.
- **Windows BSOD:** Critical system conditions result in blue screen.

## 4. Patterns in Conditional Statements

### 4.1 Simple if

```java
if (age > 18) {
    System.out.println("Adult");
}
```

### 4.2 if-else

```java
if (number % 2 == 0)
    System.out.println("Even");
else
    System.out.println("Odd");
```

### 4.3 if-else-if-else

```java
if (marks >= 90)
    System.out.println("Grade A");
else if (marks >= 75)
    System.out.println("Grade B");
else
    System.out.println("Grade C");
```

### 4.4 Nested if

```java
if (passInternal) {
    if (cgpa >= 8)
        System.out.println("Scholarship Granted");
}
```

### 4.5 switch-case

```java
switch(day) {
    case 1: System.out.println("Monday"); break;
    case 2: System.out.println("Tuesday"); break;
    default: System.out.println("Invalid Day");
}
```

## 5. Clean Coding Practices

- Use meaningful variable names.
- Avoid deep nesting.
- Return early to improve readability.
- Use logical grouping for related checks.

## 6. Common Bug: = vs == in C

```java
if (x = 5) // Bug: assignment not comparison
if (x == 5) // Correct: comparison
```

# 7. Real-World Bugs

## Amazon Shopping Cart Bug

```java
if(cartItems.size() >= 0) // Incorrect: always true
```

Should be:

```java
if(cartItems.size() > 0)
```

## Windows Blue Screen (BSOD)

Triggered by hardware failures or improper checks in kernel logic.

# 8. Practice Java Programs

## Even or Odd

```java
public static void checkEvenOdd(int n) {
    if (n % 2 == 0)
        System.out.println("Even");
    else
        System.out.println("Odd");
}
```

## Teenager Check

```java
public static void checkTeenager(int age) {
    if (age >= 13 && age <= 19)
        System.out.println("Teenager");
    else
        System.out.println("Not Teenager");
}
```

## Maximum of Three Numbers

```java
public static int maxOfThree(int a, int b, int c) {
    if (a >= b && a >= c)
        return a;
    else if (b >= c)
        return b;
    else
        return c;
}
```

## Driving License Eligibility

```java
public static void checkEligibility(int age, int vision) {
    if (age >= 18 && vision >= 7)
        System.out.println("Eligible");
    else
        System.out.println("Not Eligible");
}
```

## Hiring Rule

```java
public static void isSelected(int code, int ps, int fund, int comm, double cgpa) {
    if (code >= 4 && ps >= 4 && fund >= 4 && comm >= 3 && cgpa >= 7)
        System.out.println("Selected");
    else
        System.out.println("Rejected");
}
```

## Divisibility Check

```java
public static void checkDivisibility(int n) {
    if (n % 2 == 0 && n % 3 == 0)
        System.out.println("Divisible by both");
    else if (n % 2 == 0)
        System.out.println("Divisible by 2");
    else if (n % 3 == 0)
        System.out.println("Divisible by 3");
    else
        System.out.println("Not divisible");
}
```

## Range Check

```java
public static void checkRange(int num) {
    if (num >= 10 && num <= 20)
        System.out.println("In range");
    else
        System.out.println("Out of range");
}
```

## Nested Scholarship Check

```java
public static void checkScholarship(boolean internal, double cgpa) {
    if (internal) {
        if (cgpa >= 8)
            System.out.println("Scholarship Granted");
    }
}
```

## Merit Check

```java
public static void checkMerit(int eng, int math) {
    if (eng >= 90 || math >= 90)
        System.out.println("Merit");
}
```

## Boolean Condition Check

```java
public static void checkAllowed(boolean eligible) {
    if (!eligible)
        System.out.println("Not Allowed");
    else
        System.out.println("Allowed");
}
```