🔁 Java Loops

Loops are one of the most powerful concepts in programming. They allow a computer to repeat instructions many times without rewriting code.

Think of loops like telling a student:

> "Write your name 10 times."

Instead of repeating the instruction 10 times, you give **one instruction + a rule**.

🎯 Why Loops Are Important

Without loops:

- Programs would be very long.
- Repetitive tasks would require duplicate code.
- Maintenance would be difficult.

With loops:

- Code becomes shorter.
- Tasks become automated.
- Programs become efficient.

📌 The 3 Essential Parts of Every Loop

Every loop must have these three parts:

Step	Name	Purpose	Example
1	Initialization	Starting value	`int i = 0;`
2	Condition	When to stop	`i < 5`
3	Update	Change value	`i++`

⚠ **If update is missing → Infinite loop**

🔹 WHILE Loop (Condition-Based Loop)

👉 Use a `while` loop when you **don't know how many times** the loop should run.

Syntax

```
while(condition){
    // code
}
```

Example 1 — Print Numbers 1–5

```
int i = 1;

while(i <= 5){
    System.out.println(i);
    i++;
}
```

Output

```
1
2
3
4
5
```

Example 2 — Password Attempts

```
int attempts = 1;

while(attempts <= 3){
    System.out.println("Try password");
    attempts++;
}
```

Real-life logic:

> Keep asking until attempts reach limit.

Example 3 — Infinite Loop (Common Mistake)

```
int i = 1;
while(i <= 5){
    System.out.println(i);
}
```

❌ Problem: No `i++` → condition never changes.

◇ FOR Loop (Counting Loop)

👉 Use a `for` loop when you **know exactly how many times** you want to repeat something.

Syntax

```
for(initialization; condition; update){
    // code
}
```

Example 1 — Print 1–5

```
for(int i = 1; i <= 5; i++){
    System.out.println(i);
}
```

Example 2 — Print Even Numbers

```
for(int i = 2; i <= 10; i += 2){
    System.out.println(i);
}
```

Example 3 — Countdown Timer

```
for(int i = 5; i >= 1; i--){
    System.out.println(i);
}
System.out.println("Go!");
```

🔄 WHILE vs FOR (When to Use Which?)

Situation	Best Loop
Number of repetitions known	`for`
Unknown repetitions	`while`
Condition-based repetition	`while`
Counting numbers	`for`

👉 Rule of thumb for beginners:

> If counting → use **for** If waiting for condition → use **while**

🧠 Practice Tasks

🟢 Easy Tasks

Task 1 — While Loop Print numbers from 10 to 1.

Task 2 — For Loop Print your name 5 times.

Task 3 — For Loop Print all odd numbers from 1–15.

🟡 Medium Tasks

Task 4 — While Loop Calculate sum of numbers 1–10.

Expected Output:

```
Sum = 55
```

Task 5 — For Loop Print multiplication table of 7.

Task 6 — While Loop Print numbers divisible by 3 between 1 and 20.

🔴 Challenging Tasks

Task 7 — Pattern Printing

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

Task 8 — Reverse Counting Print numbers from 50 to 0 with step 5.

Task 9 — Guessing Game Logic Keep asking user for number until they enter 7.

(Hint: use `while` loop)

▧ Related Topics

- Type Casting
- Conditional Statements