

Loops are used to repeat certain statements based on a condition.

The following are the loops available in java:

- i) for loop
- ii) for-each loop [Enhanced for loop]
- iii) while loop
- iv) do-while loop

for-each loop:

- Also called as enhanced for loop
- Using a for-each loop we can traverse the array or collection in java.
- It is known as the for-each loop because it traverses each element one by one.
- For-each loop is not dependent on index. We can travel/print array elements without knowing the array index.

Drawbacks:

- Cannot be used to modify the array elements
- Cannot travel in reverse direction.

Syntax:

The syntax of Java for-each loop consists of data_type with the variable followed by a colon (:), then array or collection.

```
for(data_type variable : array | collection)
{
    //body of for-each loop
}
```

//Java program to display array elements using for each loop

```
class foreachdemo
{
    public static void main(String args[])
    {
        int a[]={10,20,30,40,50};
        for(int i:a)
            System.out.print(i+" ");
    }
}
```

Output:

C:\Windows\System32\cmd.exe

```
C:\Users\Admin\Desktop\Lab>javac foreachdemo.java
```

```
C:\Users\Admin\Desktop\Lab>java foreachdemo
```

```
10 20 30 40 50
```

for loop:

Syntax:

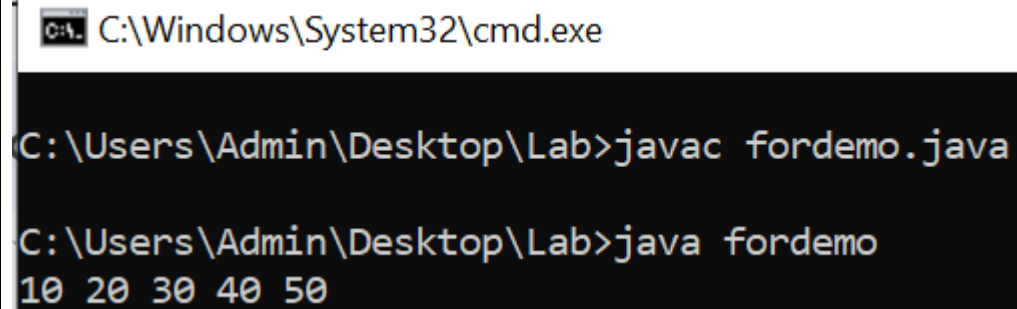
```
for(initialization;test_condition;increment/decrement)
{
    //body of the loop
}
```

//Java program to display array elements using **for loop**

```
class fordemo
{
    public static void main(String args[])
    {
```

```
int a[]={10,20,30,40,50};  
for(int i=0;i<a.length;i++)  
    System.out.print(a[i]+" ");  
}  
}
```

Output:



C:\Windows\System32\cmd.exe

```
C:\Users\Admin\Desktop\Lab>javac fordemo.java  
  
C:\Users\Admin\Desktop\Lab>java fordemo  
10 20 30 40 50
```

Differences between for and for-each loop

Aspect	For Loop	For-Each Loop
Usage	Iterates over a sequence (e.g., array, list) with full control over index and steps.	Primarily used to iterate over elements of a collection directly.
Control	You control the starting point, condition, and increment/decrement (e.g., <code>i++</code> , <code>i--</code>).	No control over index or steps, iterates over each element sequentially.
Index Access	You have access to the index of the current element (<code>array[i]</code>).	Does not provide index access, directly accesses the element.
Modifying Collection	You can modify the collection or specific elements at an index.	Not suited for modifying the collection during iteration.
Best Used For	When you need to manipulate the index or need custom increments/decrements.	When you simply need to iterate over all elements without concern for index.
Performance	May have overhead if collection size is not known and bounds need to be checked manually.	Generally faster as it doesn't require managing an index.
Example (Java)	<pre>for(int i = 0; i < arr.length; i++) { System.out.println(arr[i]); }</pre>	<pre>for(int element : arr) { System.out.println(element) ; }</pre>