**Selection Statement / Decision Making statements**

* if else
* switch

# **Iterative Statement / Loop Statement**

* while
* do while
* for
* for each (enhanced for loop)

# **Transfer Statement**

* break
* continue
* return
* try catch finally

## for loop (basic for loop)

for (*initialization*; *termination*; *increment*) {

*statement(s)*

}

When using this version of the for statement, keep in mind that:

* The ***initialization*** expression initializes the loop; it's executed once, as the loop begins.
* The boolean expression (conditional test) When the ***termination*** expression evaluates to false, the loop terminates.
* The ***increment*** expression is invoked after each iteration through the loop; it is perfectly acceptable for this expression to increment *or* decrement a value.

The three for declaration parts are separated by semicolons.

for (/\*Initialization\*/ ; /\*Condition\*/ ; /\* Iteration \*/)

{

/\* loop body \*/

}

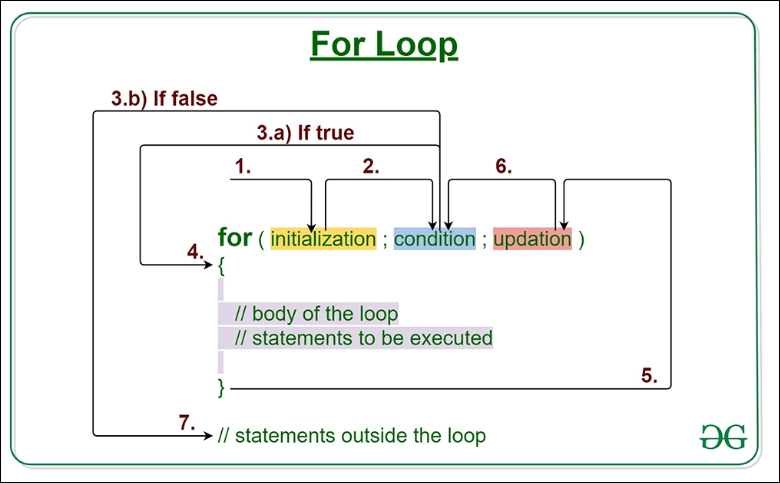
The three expressions of the for loop are optional; an infinite loop can be created as follows:

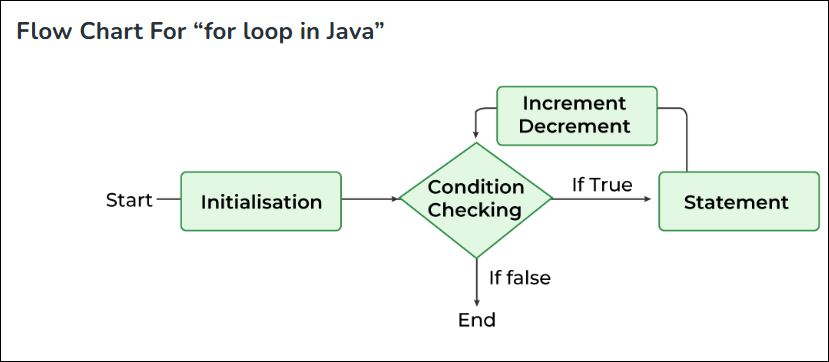
// infinite loop

for ( ; ; ) {

// your code goes here

}





### Example 1: Display a Text Five Times

// Program to print a text 5 times

class Main {

public static void main(String[] args) {

int n = 5;

// for loop

for (int i = 1; i <= n; ++i) {

System.out.println("Java is fun");

}

}

}

[Run Code](https://www.programiz.com/java-programming/online-compiler)

**Output**

Java is fun

Java is fun

Java is fun

Java is fun

Java is fun

Here is how this program works.

|  |  |  |  |
| --- | --- | --- | --- |
| Iteration | Variable | Condition: i <= n | Action |
| 1st | i = 1 n = 5 | true | Java is fun is printed. i is increased to **2**. |
| 2nd | i = 2 n = 5 | true | Java is fun is printed. i is increased to **3**. |
| 3rd | i = 3 n = 5 | true | Java is fun is printed. i is increased to **4**. |
| 4th | i = 4 n = 5 | true | Java is fun is printed. i is increased to **5**. |
| 5th | i = 5 n = 5 | true | Java is fun is printed. i is increased to **6**. |
| 6th | i = 6 n = 5 | false | The loop is terminated. |

The Basic for Loop: **Declaration and Initialization**: you declare and initialize zero, one, or multiple variables of the same type inside the parentheses after the for keyword. If you declare more than one variable of the same type, then you'll need to separate them with commas as follows:

for (int x = 10, y = 3; y > 3; y++) {

}

|  |  |
| --- | --- |
| for (int x = 1; x < 2; x++) {  System.out.println(x); // Legal  }  System.out.println(x); // Not Legal! x is now out of scope // and can't be accessed. |  |

Basic for Loop: **Conditional (boolean) Expression :** must evaluate to a boolean value

for (int x = 0; ((((x < 10) && (y-- > 2)) | x == 3)); x++) { }

Basic for Loop: **Iteration Expression :** After each execution of the body of the for loop, the iteration expression is executed.

|  |  |
| --- | --- |
| for (int x = 0; x < 3; x++) {  System.out.println("in for loop");  return true;  } | return Execution jumps immediately back to the calling method. |
| break; | break Execution jumps immediately to the 1st statement after the for loop. |
| System.exit() | All program execution stops; the VM shuts down. |
| for Loop Issues | for( ; ; ) {  System.out.println("Inside an endless loop");  } |
| int i = 0; for (;i<10;){  i++;  } |  |
| for (int i = 0,j = 0; (i<10)&&(j<10); i++,j++){  } | Multiple variables. |
| int x = 3; for (x = 12; x < 20; x++) { } System.out.println(x); | Initialize value |
| int b = 3; for (int a = 1; b != 1; System.out.println("iterate"))  {  b = b - a;  } | iterate  iterate |

## **Nested Loops**

// Outer loop

for (int i = 1; i <= 2; i++) {

System.out.println("Outer: " + i); // Executes 2 times

// Inner loop

for (int j = 1; j <= 3; j++) {

System.out.println(" Inner: " + j); // Executes 6 times (2 \* 3)

}

}

**Print table from 1 to 10:**

for(int i=1; i<=10;i++){

for(int j=1; j<=10;j++){

System.out.print(i\*j+" ");

}

System.out.println("");

}

**For each loop for arrays and collections.**

int [] a = {1,2,3,4};

for(int x = 0; x < a.length; x++)

// basic for loop

System.out.print(a[x]);

For-each loop (enhanced for loop came from java 1.5 version)

----------------------------------------------------------------------------------

Specially designed for loop to retrieve elements for arrays and collections.

for(int n : a){

// enhanced for loop

System.out.print(n);

}

More formally, let's describe the enhanced for as follows:

for(declaration : expression)

The two pieces of the for statement are

■ **Declaration** The newly declared block variable, of a type compatible with the elements of the array you are accessing. This variable will be available within the for block, and its value will be the same as the current array element.

■ **expression** This must evaluate to the array you want to loop through. This could be an array variable or a method call that returns an array. The array can be any type: primitives, objects, even arrays of arrays.

For loop is index based.

And for each loop in value based.

int x;

long x2;

Long [] La = {4L, 5L, 6L};

long [] la = {7L, 8L, 9L};

int [][] twoDee = {{1,2,3}, {4,5,6}, {7,8,9}};

String [] sNums = {"one", "two", "three"};

Animal [] animals = {new Dog(), new Cat()};

// legal 'for' declarations

for(long y : la ) ;

// loop thru an array of longs for(long lp : La) ;

// autoboxing the Long objects

// into longs

for(int[] n : twoDee) ; // loop thru the array of arrays

for(int n2 : twoDee[2]) ; // loop thru the 3rd sub-array

for(String s : sNums) ; // loop thru the array of Strings

for(Object o : sNums) ; // set an Object reference to // each String

for(Animal a : animals) ; // set an Animal reference to each // element

* Curly brace is not mandatory
* cannot iterate in reverse iteration, support only in forward direction.
* Cannot skip in the iteration. Like skip i=i+3
* Go for, all element in forward direction iteration
* Cannot iterate fix number of type, print 10 times "RAVI"

## What is the Difference Between for loop and foreach Loop?

|  |  |
| --- | --- |
| for Loop vs foreach Loop | |
| The for loop is a control structure for specifying iteration that allows code to be repeatedly executed. | The foreach loop is a control structure for traversing items in an array or a collection. |
| **Element Retrieving** | |
| A for loop can be used to retrieve a particular set of elements. | The foreach loop cannot be used to retrieve a particular set of elements. |
| **Readability** | |
| The for loop is harder to read and write than the foreach loop. | The foreach loop is easier to read and write than the for loop. |
| **Usage** | |
| The for loop is used as a general purpose loop. | The foreach loop is used for arrays and collections. |

**Assignments: write the output**

|  |  |
| --- | --- |
| class Test {      public static void main(String[] args)      {          for (int i = 1; i <= 10; i++) {              System.out.println(i);          }      }  } | **1**  **2**  **3**  **4**  **5**  **6**  **7**  **8**  **9**  **10** |
| class forLoopDemo {      public static void main(String args[])      {          for (int i = 1; i <= 5; i++)              System.out.println("Hello World");      }  } | **Hello World**  **Hello World**  **Hello World**  **Hello World**  **Hello World** |
| class forLoopDemo {      public static void main(String args[])      {          int sum = 0;          for (int x = 1; x <= 20; x++) {              sum = sum + x;          }          System.out.println("Sum: " + sum);      }  } | **???** |
| public class enhancedforloop {        public static void main(String args[])      {          String array[] = { "Ron", "Harry", "Hermoine" };          for (String x : array) {              System.out.println(x);          }            /\* for loop for same function          for (int i = 0; i < array.length; i++)          {              System.out.println(array[i]);          }          \*/      }  }  class Test {      public static void main(String args[])      {          for (int i = 1; i >= 1; i++) {              System.out.println("Stop me " + i);          }      }  } |  |
| public class Test{        public static void main(String[] args)      {          for (;;) {              System.out.println("loop");          }      }  } |  |
| Write table from 1 to 10 |  |
| Print number in reverse order |  |
| Print even number up to 100 |  |
| Print odd number up to 100 |  |
| Print below pattern  \*  \*\*  \*\*\*  \*\*\*\*  \*\*\*\*\* |  |
| Print below pattern  1  12  123  1234  12345 |  |
| Use nested for loop and break statement and verify which for loop breaks. |  |
| Use labeled statement in for loop with break, continue |  |
| Use return keyword in for loop |  |
| What is System.exit(0) |  |