

Java Switch Statement :-

The switch statement allows us to execute a block of code among many alternatives.

The syntax of the switch statement in java -

Syntax -

```
switch (Expression) {
```

```
    case value 1 :
```

```
        // Code
```

```
        break ;
```

```
    case value 2 :
```

```
        // Code
```

```
        break ;
```

```
    ...
```

```
    default :
```

```
        // default statements
```

```
}
```

Example of java switch statement -

```
// java Program to check the size  
// using the switch ... case statement
```

```
class main
```

```
{
```

```
Public static void main (String [] args)
```

```
{
```

```
    int number = 44;
```

```
    String size;
```

```
    // Switch statement to check size
```

```
    switch (number) {
```

```
        case 29 :
```

```
            size = "Small";
```

```
            break;
```

```
        case 42 :
```

```
            size = "Medium";
```

```
            break;
```

```
    // match the value of weak
```

```
        case 44 :
```

```
            size = "Large";
```

```
            break;
```

```
        case 48 :
```

```
            size = "Extra large";
```

```
            break;
```

```
        default :
```

```
            size = "Unknown";
```

```
            break;
```

```
    }
```

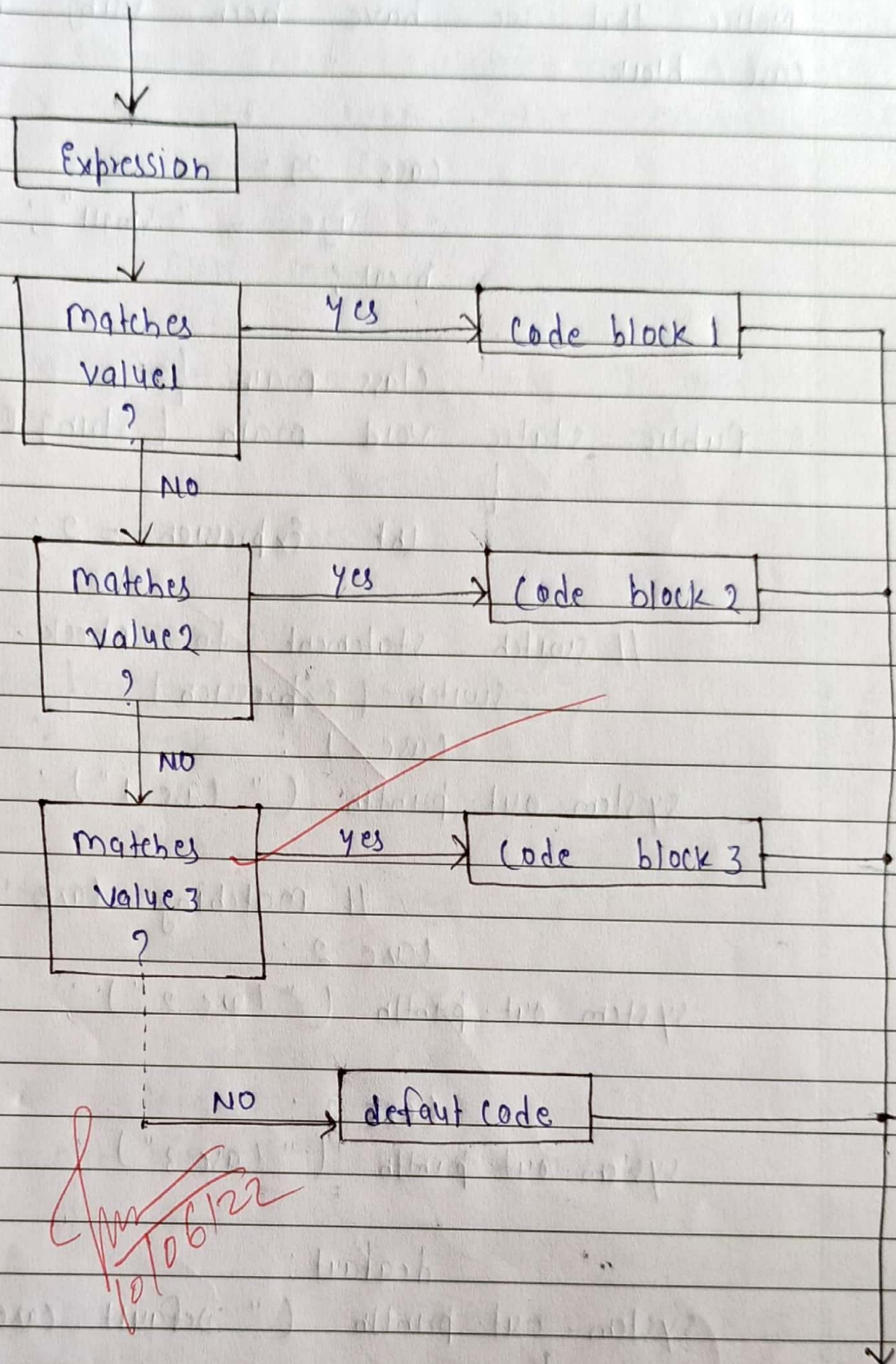
```
    System.out.println ("Size: " + size);
```

```
}
```

```
}
```

O/P - Size: Large

flowchart of switch statement :-



Signature
10/06/22

switch terminates

Break statement in java Switch ... Case -

Notice that we have been using break in each case block.

```
.....
    case 29 :
        size = "small";
        break;

class main {
    public static void main (String [] args)
    {
        int Expression = 2;

        // Switch statement to check size
        switch (Expression) {
            case 1 :
                system.out.println ("Case 1");

                // matching case
            case 2 :
                system.out.println ("Case 2");

            case 3 :
                system.out.println ("Case 3");

            default :
                system.out.println ("default case");
        }
    }
}
```

Output - Case 2
Case 3
default case

Default Case in java switch - case :-

The switch statement also includes an optional default case. It is executed when the expression doesn't match any of the cases.

Example -

```
class main
{
    public static void main ( String [] args)
    {
        int expression = 9;
        switch (expression) {
```

Case 2 :

```
    system.out.println (" Small Size" );
    break;
```

Case 3 :

```
    system.out.println (" Large Size" );
    break;
```

// default case

default :

```
    system.out.println (" Unknow Size" );
    }
}
```

Output - Unknow Size

Java for loop :-

In Computer Programming, loops are used to repeat a block of code. For example, if you want to show a message 100 times, then rather than typing the same code 100 times, you can use a loop.

In java there are three types of loops-

1. for loop
2. while loop
3. do while loop

1. Java for loop :-

Java for loop is used to run a block of code for a certain number of times. The syntax of for loop is-

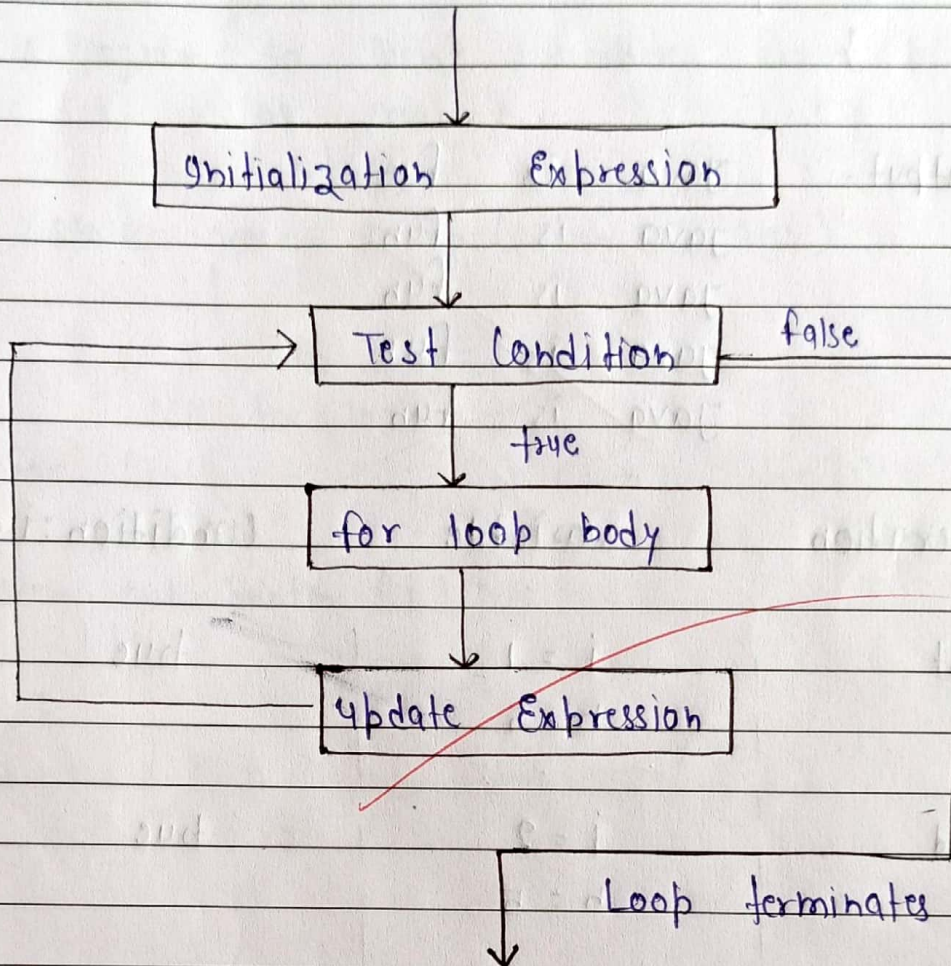
Syntax -

```
for (initialExpression; testExpression; updateExpression)
{
    // body of the loop
}
```

Here,

The initial Expression initializes and / or declares and executes only once.

flowchart of java for loop:-



Example 1. of 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");
  }
}

```

output-
 java is fun
 java is fun
 java is fun
 java is fun
 java is fun

Iteration	variable	Condition: $i \leq n$	Action
1st	$i = 1$ $n = 5$	true	java is fun is printed
2nd	$i = 2$ $n = 5$	true	"
3rd	$i = 3$ $n = 5$	true	"
4th	$i = 4$ $n = 5$	true	"
5th	$i = 5$ $n = 5$	true	"
6th	$i = 6$ $n = 5$	false	The loop is terminated.

Example 2 - Display numbers from 1 to 5 $\Rightarrow$

```
// Program to Print numbers from 1 to 5
class main
{
    public static void main (String[] args)
    {
        int n = 5;
        // for loop
        for (int i = 1; i <= n; ++i)
        {
            System.out.println (i);
        }
    }
}
```

output - 1

2

3

4

5.

Example 3 - Display Sum of n Natural Numbers $\Rightarrow$

// Program to find the sum of Natural numbers from 1 to 100
class main

```
public static void main (String[] args)
{
    int sum = 0;
```

```

int n = 1000;
// for loop
for (int i = 1; i <= n; ++i)
{
    // body inside for loop
    sum += i;
    // sum = sum + i
}
System.out.println("sum = " + sum);
}

```

output - 500500

Java for - Each loop :-

The java for loop has an alternative syntax that makes it easy to iterate through arrays and collections.

Example -

```

// print array element
class main
{
    public static void main (String [] args)
    {
        // create an array
        int [] numbers = {3, 7, 5, -5};

        // iterating through the array
        for (int number : numbers)
        {
            System.out.println (number);
        }
    }
}

```

O/P -
3
7
5
-5

Java Infinite for loop :-

if we set the test expression in such a way that it never evaluates to false, the for loop will run forever. This is called infinite for loop

Example -

```
// Infinite for loop
class Infinite
{
    public static void main (String [] args)
    {
        int sum = 0;
        for (int i = 1; i <= 10; --i)
        {
            System.out.println ("Hello");
        }
    }
}
```

Output -

Hello

Hello

Hello

Hello

Hello

Hello

Hello

Hello

⋮

∞ (Infinite) (31-11-22)

Java for - Each loop :-

In java, the for - Each loop is used to iterate through elements of arrays and collections (like ArrayList). It is also known as the Enhanced for loop.

for- Each Loop Syntax -

```
for (datatype item : array)
{
    .....
}
```

Example of Print Array Elements -

```
// Print array elements
class main {
    public static void main (String [] args)
    {
        // create an array
        int [] numbers = { 3, 9, 5, -5 };

        // for each loop
        for (int number : numbers)
        {
            System.out.println (number);
        }
    }
}
```

output -
3
9
5
-5

for loop vs for - Each loop :-

Let's see how a for - Each loop is different from a Regular java for loop.

1. Using for loop :-

```
class main
{
    public static void main (String [] args)
    {
        char [] vowels = { 'a', 'e', 'i', 'o', 'u' };

        // iterating through an array using a for loop
        for (int i = 0; i < vowels.length; ++i)
        {
            System.out.println (vowels [i]);
        }
    }
}
```

output -
a
e
i
o
u

2. Using for - Each loop :-

```

class main
{
    public static void main (String [] args)
    {
        char [] vowels = {'a', 'e', 'i', 'o', 'u'};

```

// iterating through an array using the for-each loop

```

for (char item : vowels)
{

```

```

    System.out.println (item);
}
}

```

Output -

a

e

i

o

u

2. Java while loop :-

Java while loop is used to run a specific code until a certain condition is met.

Syntax-

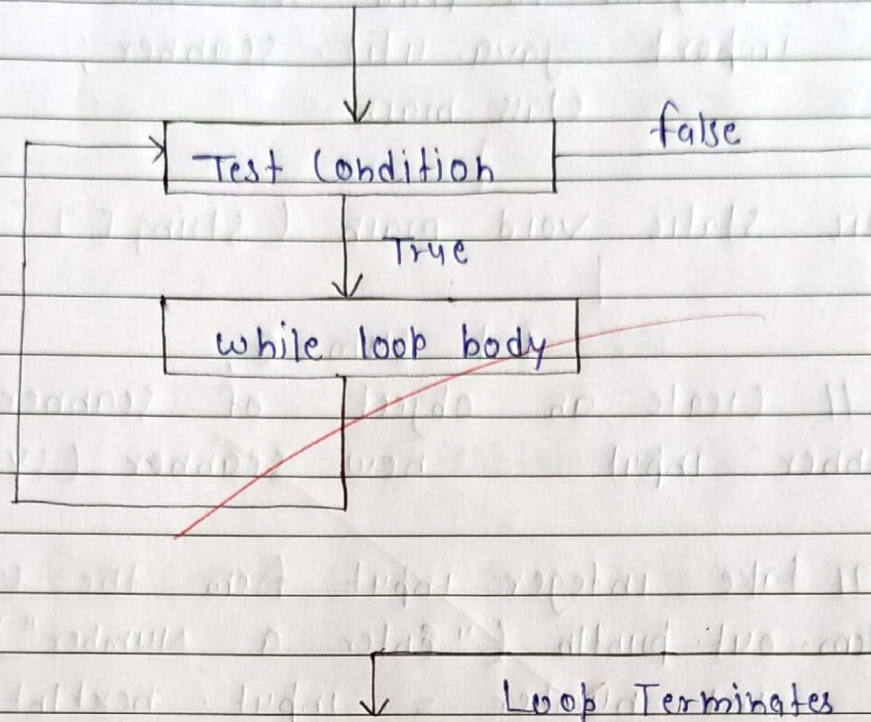
```

while (testExpression) {
    // body for loop
}

```

Here,

- A while loop evaluates the testExpression inside the Parenthesis.
- if the testExpression evaluates to true, the code inside the while loop is executed.
- The testExpression is evaluated again.

flowchart of while loop :-

Example 1- Display Number from 1 to 5 -

// Program to display Numbers from 1 to 5

class main

{

Public static void main (String [] args)

{

// declare variables

int i = 1, n = 5;

// while loop from 1 to 5

while (i <= n) {

System.out.println (i);

i++

}

}

}

output - 1

2

3

4

5

Example 2 - Sum of Positive Numbers only -

```
// Java Program to find the Sum of Positive Numbers
import java.util.Scanner;
class main
{
    public static void main (String[] args)
    {
        int sum = 0;
        // Create an object of Scanner class
        Scanner input = new Scanner (System.in);

        // take integer input from the user
        System.out.println ("Enter a Number");
        int number = input.nextInt();

        // while loop continues
        // until entered number is Positive
        while (number >= 0)
        {
            // add only positive numbers
            sum += number;
            System.out.println ("Enter a number");
            number = input.nextInt();
        }
        System.out.println ("Sum = " + sum);
        input.close();
    }
}
```

Output →	Enter a number, 25
	Enter a number, 9
	Enter a number, 5
	Enter a number, -3
	Sum = 39

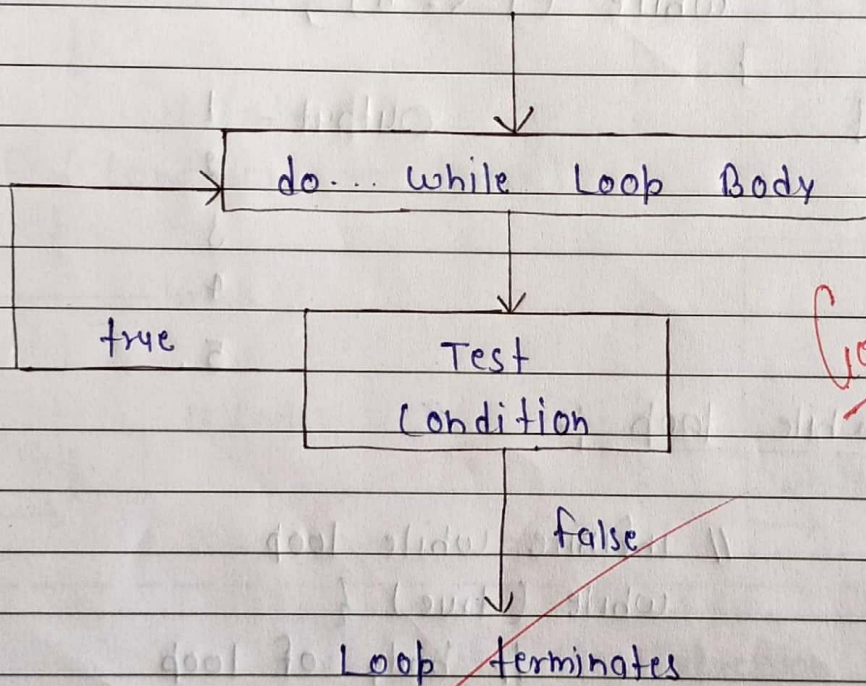
3. Java do... while loop :-

The do... while loop is similar to while loop. However, the body of do... while loop is executed once before the test expression is checked.

Ex-

```
do {  
    // body of loop  
}  
while (testExpression);
```

flowchart of do... while loop :-



Good
10/06/22

Example of Display Numbers from 1 to 5:-

// java Program to display numbers from 1 to 5

```
import java.util.Scanner;
```

// Program to find the Sum of Natural numbers
// 1 to 100.

```
class main
{
    public static void main (String[] args)
    {
        int i = 1, n = 5;
        // do... while loop from 1 to 5
        do
        {
            System.out.println(i);
            i++;
        }
        while (i <= n);
    }
}
```

Output -

1

2

3

4

5

Infinite while loop

E.g.

```
// infinite while loop
while (true) {
    // body of loop
}
```

Here, is an example of an infinite do... while loop.

```
// infinite do... while loop
```

```
int count = 1;
do {
    // body of loop
} while (count == 1)
```