**Accessing static and non static methods**

#### In both static and non-static methods, static methods are directly accessed.

import java.io.\*;

public class Static\_method {

static int num = 100;

static String str = "Dr.Sm.Badhusha";

// This is Static method

static void display()

{

System.out.println("static number is " + num);

System.out.println("static string is " + str);

}

// non-static method

void nonstatic()

{

// our static method can be accessed

// in non static method

display();

}

// main method

public static void main(String args[])

{

Static\_method obj = new Static\_method();

// This is object to call non static function

obj.nonstatic();

// static method can called

// directly without an object

display();

}

}

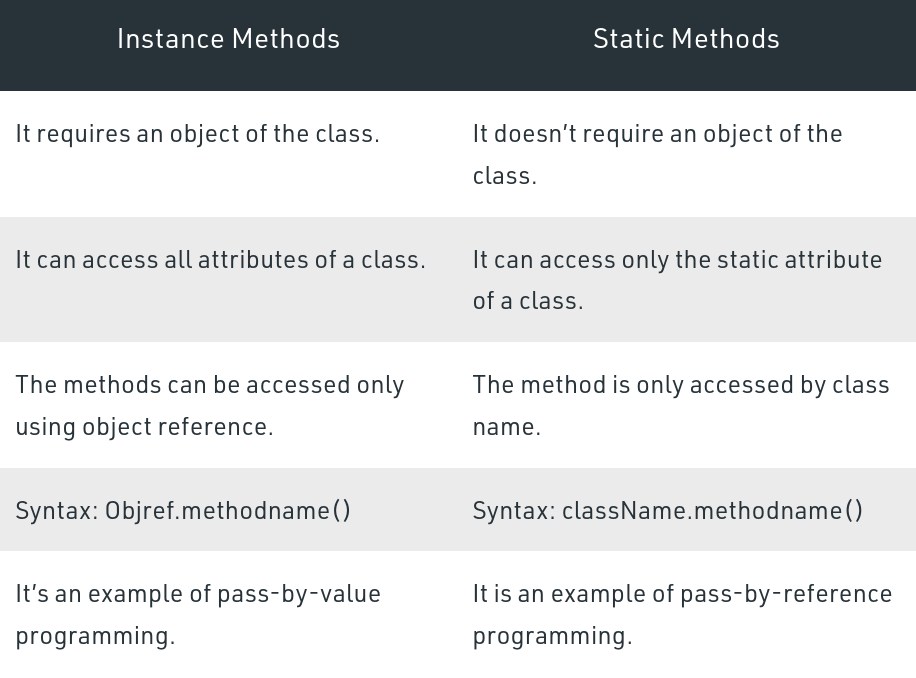
o/p

static number is 100

static string is Dr.Sm.Badhusha

static number is 100

static string is Dr.Sm.Badhusha



TheMath class in Java has almost all of its members static. So, we can access its members without creating instances of the Math class. For example,

public class Math\_class {

public static void main( String[] args ) {

// accessing the methods of the Math class

System.out.println("Absolute value of -12 = " + Math.abs(-12));

System.out.println("Value of PI = " + Math.PI);

System.out.println("Value of E = " + Math.E);

System.out.println("2^2 = " + Math.pow(2,2));

System.out.println("Square root of 5 is "+Math.sqrt(5));

}

}

o/p

Absolute value of -12 = 12

Value of PI = 3.141592653589793

Value of E = 2.718281828459045

2^2 = 4.0

Square root of 5 is 2.23606797749979

In the above example, we have not created any instances of the Math class. But we are able to access its methods: abs() and pow() and variables: PI and E.

It is possible because the methods and variables of the Math class are static.