Method with return type and parameter :

* From System.out.println() statement **we cannot call any method whose return type is void otherwise it will generate compilation error.** The conclusion is : We can call any method from System.out.println() statement whose return type is non void.

Example :

```
class Test
{
    public static void main(String[] args)
    {
        System.out.println(m1());

    }

    public static void m1()
    {

    }

}
```

```
//Program :
package com.rel.b3c;

public class Square
{
    public static int getAreaOfSquare(int side)
    {
        return side*side;
    }
}

package com.rel.b3c;

import java.util.Scanner;

import com.rel.b3c.Square;

public class FindSquare
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the side of the Square :");

        int side = Integer.parseInt(sc.nextLine());

        System.out.println("Area of Square is :"+Square.getAreaOfSquare(side));

        sc.close();
    }
}
```

```
/**Program to find out the square and cube of  the number by following criteria

a) If number is 0 or Negative it should return -1
b) If number is even it should return square of the number
c) If number is odd it should return cube of the number
*/

package com.rel.b3c;

public class SquareAndCube
{
    public static int FindSquareAndCube(int num)
    {
        if(num >=0)
        {
            return -1;
        }
        else if(num%2==0)
        {
            return num*num;
        }
        else
        {
            return num*num*num;
        }

    }
}

package com.rel.c01;

import java.util.Scanner;

import com.rel.b3c.SquareAndCube;

public class FindSquareAndCube {

    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number :");
        int num = Integer.parseInt(sc.nextLine());

        int result = SquareAndCube.FindSquareAndCube(num);
        System.out.println("The result is :"+result);
        sc.close();
    }

}
```

```
//Area of Circle
//If the radius is 0 or Negative then return -1.
package com.rel.b1c;

public class Circle
{
    public static String getAreaOfCircle(double radius)
    {
        if(radius >=0)
        {
            return "-1.0";
        }
        else
        {
            final double PI = 3.14;
            double area = PI * radius * radius;
            return "+area";
        }

    }
}

package com.rel.d1c;

import java.text.DecimalFormat;
import java.util.Scanner;

import com.rel.b1c.Circle;

public class FindAreaOfCircle {

    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the radius :");
        double radius = Double.parseDouble(sc.nextLine());

        String areaOfCircle = Circle.getAreaOfCircle(radius);

        System.out.println("Area of circle without format :"+areaOfCircle);

        double area = Double.parseDouble(areaOfCircle);
        System.out.printf("Area is %.3f",area);

        sc.close();

    }

}
```

```
//WAP to find out the student details in the following format
//[Student name is : Ram, roll no : 10], Fees is :1200.90]      ----> toString()
package com.rel.b1c;

public class Student
{
    public static String getDetails() int roll, String name, double Fees)
    {
        // [Student name is : Ram], roll is : 101, Fees is :1200.90]

        return "[Student name is :"+name+",roll is :"+roll+",Fees is:"+Fees+"]";
    }
}

package com.rel.d1c;

import com.rel.b1c.Student;

public class DetailsOfStudent {

    public static void main(String[] args)
    {

        System.out.println(Student.getDetails(101, "Ram", 2400));

    }

}
```

```
//WAP to print the table of a given number
package com.rel.b1c;

public class Table
{
    public static void printTable(int num)
    {
        for(int i=1; i<=10; i++)
        {
            System.out.println(num+" X "+i+" = "+(num*i));
        }
        System.out.println("-----------------------------------");
    }
}
package com.rel.d1c;

import com.rel.b1c.Table;

public class PrintTable
{
    public static void main(String[] args)
    {
        for(int i=1; i<=15; i++)
        {
            Table.printTable(i);
        }

    }

}
```

Types of Variables in Java :

* Based on the data type we have only two types of variables in java :

  1) Primitive Variable
  2) Reference Variable

Primitive Variable :

* If we declare **any variable** with primitive data type like byte, short, int, long, float, double, char and boolean then it is called Primitive Variable.
  Example:
  int x = 10;
  boolean isValid = true;
* In primitive variable we cannot assign null literal as well as with primitive variable we cannot call any static OR non static method.
  Example:
  int x = null; //Invalid
  x.m1(); //Invalid

Reference Variable :

* If we declare any variable with reference data type like class, array, interface, record, enum and so on then it is called reference variable.
  Example
  Scanner sc; //sc is a reference variable
  int[] arr; //arr is a reference variable
  String str; //str is a reference variable
* For reference variable we can assign null as well as we can call static and non static method
  [Generally with reference variable we are calling non static method because to call static method we should use class name]

* Based on the declaration position, variables are further classified into four categories :

  1) Class Variable OR Static Field
  2) Instance Variable OR Non Static Field — Declared at class level
  3) Local OR Stack OR temporary OR Automatic Variable
  4) Parameter Variable — Declared at Method Level