

```
//Working with static method with different return type :
package com.ravi.blc;

//BLC
public class Student
{
    public static String getDetails(int roll, String name, String address)
    {
        // [Student id is : 101, Name is : Scott, Address is : Ameerpet]
        return "[Student id is :" + roll + ", Name is :" + name + ", Address is :" + address + "]";
    }
}

package com.ravi.elc;
import com.ravi.blc.Student;
public class Test4 {

    public static void main(String[] args)
    {
        String details = Student.getDetails(101, "Scott", "Ameerpet");
        System.out.println(details);
    }
}

package com.ravi.blc;
public class EvenOrOdd
{
    public static boolean isEven(int num)
    {
        return (num%2==0);
    }
}

package com.ravi.elc;
import java.util.Scanner;
import com.ravi.blc.EvenOrOdd;
public class Test5 {

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

        System.out.print("Is " + num + " even number ? " + EvenOrOdd.isEven(num));
        sc.close();
    }
}

package com.ravi.blc;
public class Table
{
    public static void printTable(int num) // 5 X 1 = 5
    {
        for(int i=1; i<=10; i++)
        {
            System.out.println(num + " X " + i + " = " + (num*i));
        }
        System.out.println("-----");
    }
}

package com.ravi.elc;
import java.util.Scanner;
import com.ravi.blc.Table;
public class Test6 {

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

        for(int i=1; i<=num; i++)
        {
            Table.printTable(i);
        }

        sc.close();
    }
}
```

Types of variable in java :

---

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

- 1) Primitive Variables
- 2) Reference Variables

Primitive Variables :

\* If we declare any variable with **primitive data type** i.e byte, short, int, long, float, double, char and boolean then It is primitive variable.

Example : int x = 100;

\* We cannot assign **null** literal on primitive variable as well as we **cannot call any method** by using primitive variable.

int x = null; //Invalid

int y = 90;  
y.m1(); //Invalid

**Reference Variable :**

\* If a variable is declared with reference data type (class name) then it is called Reference Variable.

Example : Test t1 = new Test();

Here t1 is a reference variable.

\* We can assign null literal on reference variable as well as with the help of reference variable we can call non static method with proper object reference.

Now, Based on the **declaration position**, Variables are further classified into 4 types :

1) Class Variable OR Static Field

2) Instance Variable OR Non static Field

At Class Level

[ONLY THE CLASS LEVEL VARIABLES ARE HAVING DEFAULT VALUE]

3) Local Variable

4) Parameter Variable

At Method Level

Variables

At Class Level

Static Field      Non static Field

At Method Level

Local Variable

Parameter Variable

Example :

class Test

{

    public void accept()

    {

        int d = 400;

        System.out.println("Class Variable :" + Test.a);

        System.out.println("Instance Variable :" + b);

        System.out.println("Parameter Variable :" + c);

        System.out.println("Local Variable :" + d);

    }

}

    public class PrimitiveVariable {

        public static void main(String[] args)

        {

            Test t1 = new Test();

            t1.accept(300);

        }

}

    //Program on Primitive Variables :

    //BLC

    public class Test

    {

        static int a = 100; //class variable OR Static Field

        int b = 200; //Instance variable OR non static field

    }

    public void accept(int c)

    {

        int d = 400;

        System.out.println("Class Variable :" + Test.a);

        System.out.println("Instance Variable :" + b);

        System.out.println("Parameter Variable :" + c);

        System.out.println("Local Variable :" + d);

    }

}

    //Program on Reference Variable :

    //BLC

    import java.util.Scanner;

    class Student

    {

        Student s1 = new Student(); //s1 is non static reference variable

        static Scanner sc = new Scanner(System.in); //sc is static reference variable

    }

    public void accept(Student s2) //s2 is a parameter variable

    {

        Student s3 = new Student(); //s3 is local variable

    }

    public class ReferenceVariable

    {

        public static void main(String[] args)

        {

        }

    }

}