SHRI VILEPARLE KELAVANI MANDAL'S
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC ACCREDITED with "A" GRADE (CGPA : 3.18)

DEPARTMENT OF INFORMATION TECHNOLOGY

COURSE CODE: DJS22ITL306 **DATE:28-10-23**

COURSE NAME: Programing Laboratory 1 (Advanced Java)

CLASS: S.Y B. Tech IT

Name: Anish Sharma

EXPERIMENT NO.2

CO/LO:
CO1- Modify the behaviour of methods, classes, and interfaces at runtime.

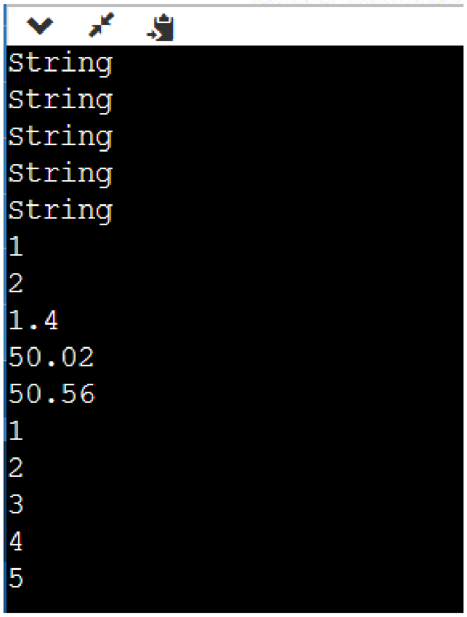
AIM / OBJECTIVE:

To implement different generic types

PROBLEM STATEMENTS:

1. Write a Java program that creates 2 lists, 1 for integer and other for strings. Define a generic method to display the elements of both lists using arrays with the use of for-each loop.

```
import java.util.*;

class Test
{
    String var;

    <A>Test(A var)
    {
        this.var = var.toString();
    }

    String display() {
        return var.getClass().getSimpleName();
    }

    void display(List<?super Integer> list) {
        for (Object i : list) {
            System.out.println(i);
        }
    }
}

public class Main
{
    public static void main(String[] args)
    {
        Test a = new Test(100);
        Test b = new Test("Amit");
        Test c = new Test(true);
        Test d = new Test(100.321);
        Test e = new Test(100.321f);
        Test f = new Test(100.321d);

        System.out.println(a.display());
        System.out.println(b.display());
        System.out.println(c.display());
        System.out.println(d.display());
        System.out.println(e.display());

        List<Number> l1 = Arrays.asList(1,2,1.4,50.02,50.56);
        List<Integer> l2 = Arrays.asList(1,2,3,4,5);
        a.display(l1);
        a.display(l2);
    }
}
```

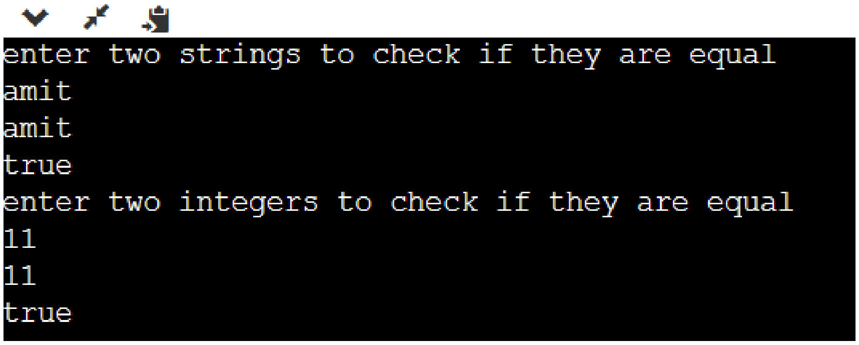


```
String
String
String
String
String
1
2
1.4
50.02
50.56
1
2
3
4
5
```

2. Write a simple generic version of method isEqualTo that compares its two arguments with the equals method and returns true if they’re equal and false otherwise. Use this generic method in a program that calls isEqualTo with a variety of built-in types, such as Object or Integer. What result do you get when you attempt to run this program?

```
    import java.util.*;
public class Main {


  public static <T> boolean isEqualTo(T first, T second) {
    return first.equals(second);
  }


  public static void main(String[] args) {
    Scanner sc=new Scanner(System.in);
    System.out.println("enter two strings to check if they are equal");
    System.out.println(isEqualTo(sc.next(),sc.next()));
    System.out.println("enter two integers to check if they are equal");
    System.out.println(isEqualTo(sc.nextInt(),sc.nextInt()));
```

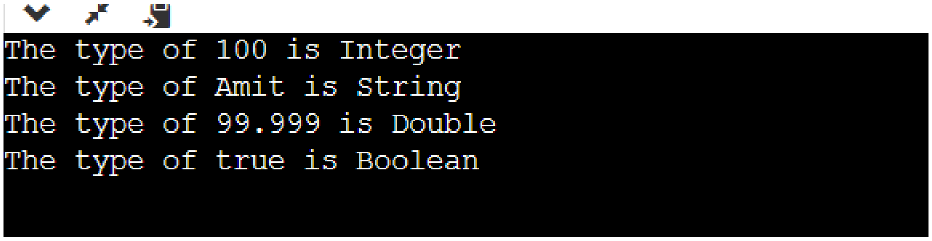
```
        }

    }
```



```
enter two strings to check if they are equal
amit
amit
true
enter two integers to check if they are equal
11
11
true
```

3. **3.** Write a generic method Sort based on the sort program. Write a test program that inputs, sorts and outputs an Integer array and a Float array. [Hint: Use > in the type-parameter section for method Sort, so that you can use method compareTo to compare the objects of the type that T represents.]

```
class Test<X,Y> {

 X var1;

 Y var2;


 Test(X var1, Y var2) {

    this.var1 = var1;

    this.var2 = var2;

 }


 Y getValue() {

    System.out.println("The type of " + var1 + " is " + var1.getClass().getSimpleName());

    System.out.println("The type of " + var2 + " is " + var2.getClass().getSimpleName());


    return (Y)var2.getClass().getSimpleName();

 }
}


public class Main {
    public static void main(String[] args) {
        Test<Integer, String> a = new Test<Integer, String>(100, "Amit");
        Test<Double, Boolean> b = new Test<Double, Boolean>(99.999, true);
        a.getValue();
        b.getValue();
    }
}
```



```
The type of 100 is Integer
The type of Amit is String
The type of 99.999 is Double
The type of true is Boolean
```

CONCLUSION:

We learnt to implement Genrics in java