LAB-07 Generic

```
import java.io.*;
import java.lang.*;
import java.util.*;

class gen <T>
{
  T ob;
  gen (T o)
  {
    ob = o;
  }
  T getob ()
  {
    return ob;
  }
  void showtype ()
  {
    System.out.println ("Type of T is " + ob.getClass ().
                                        getName ());
  }
}
Class generic
{
  public static void main (string [] args)
  {
    string n;
    Scanner sc = new Scanner (system.in);
  System.out.println ("Enter the Integer Number to be
    displayed using the generic style");
          n = sc.next ();
```

```
gen <Integer> ob1 = new gen<Integer>(Intege.
                                    praseInt(n));

ob1.showtype();
int val=ob1.getob();
System.out.println("Value is: "+val);

System.out.println();

System.out.println("Enter the string to be displayed
                    using generic style");
n=sc.next();
gen<String> ob2=new gen<String>(n);
Ob2.showtype();
String x=ob2.getob();
System.out.println("Value: "+x);

System.out.println();

System.out.println("Enter the Double Number to be
    displayed using generic style");
n=sc.next();
gen<Double> ob3 = new gen<Double>(Double.parse
                                    double(n));

ob3.showtype();
double ans = ob3.getob();
System.out.println("Value : "+ans);
  }
}
```