LAB-17 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>(Integer.
                                   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.get ob();
System.out.println("Value : "+ ans);
   }
}
```

LAB - 9

```
import.java.util.* ;
importjava.io.* ;
import java.lang.*;
class newthread implement Runnable
 {
  Thread t ;
   new thread ()
 {
    t = new thread (this, "CSE");
    System. out. println ("CHILD THREAD : " + t);
 }
~~put~~ public void run ()
 {
   try {
   for (;;)
  {
      System.out.println ("CSE");
         Thread. sleep (2000);
    }
  } catch (Interrupted Exception e) {
            System.out.println ("CSE Thread interrouoted");
  }
  System.out. println ("Exiting The CSE Thread");
   }
 }
}
 Class Thread
{
   Public static void main (string[]args]
   {
```

```
newthread nt = new thread();
  nt.t.start();
   try {
        for (;;)
        {
         System.out.println("BMS COLLEGE OF ENGINEERING");
          Thread.sleep(10000);
         }
    } catch (InterruptedException e) {
         System.out.println("Main Thread Interrupted
                                               ");
      }
     system.out.println("Existing out of the
                   main thread");
         }
      }
```

Lab-10 MULTIPLICATION THREAD

```
class Table
{
void printTable(int n)
{
 synchronized (this)
 {
  for (int i=1; i<=5; i++)
  {
   system.out.println(+n+"*"+i+"="+(n*i));
 try
   {
   Thread.sleep (400);
 }
  catch (Exception e)
 {
  system.out.println (e);
 }
}
}
}
}
class mythread1 extends Thread
 {
  Table t;
  Mythread 1 (Table t)
 {
  this. t=t;
 }
 public void run()
{
```

```
t.printTable(5)
 }
}
class My thread 2 extends Thread
{
 Table t;
 My thread 2 (Table t)
 {
  this.t=t;
 }
 public void run ()
 {
  t.printTable (100);
  }
 }
class Use
{
Public static void main (string args[])
 {
  Table obj= new Table()
   My thread 1 th 1 =new
  My thread 1 (obj)
    My thread 2 th 2 =new
   My thread 2 (obj);
    th 1. Start();
    th2. start();
   }
  }
```