27/11/20

LAB - 7

SAIPRAVEEN MARNI
1BM19CS138

Write a program to demonstrate generics with multiple object parameters.

```
        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 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 systle");
              n=sc.next();
       gen<Integer> ob1 = new gen<Integer>(Integer.parseInt(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 sytle");
              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.parseDouble
                                                        (n));

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

OUTPUT :-

```
Enter the Integer Number to Be Displayed Using generic Sytle
  10
 Type of T is java.lang.Integer
  Value is : 10
Enter the Double Number to Be Displayed using generic style
  3.14216
 Type of T is java.lang.Double
  Value is : 3.14216
```

27/11/20

LAB - 8

SAIPRAVEEN MARNI
1BM19CS138

Write a program that demonstrates handling expections in inheritance tree. Create a base class called "Father" and derived class "Son" which extends the base class.

In Father class, implement a construct which takes the age and throws the expectional WrongAge() when the input age < 0. In son class, implement a constructor that cases both father and son's age throws an expection if son's age is >= father's age.

```
    import java.util.*;
  class WrongAge extends Exception {
     int detail;
    WrongAge (int A) {
      detail = a;
  }
   public string toString () {
   return "enter correct age "+ detail" is invalid";
   }
 }
   class father {
     Public int age;
   Scanner in = new Scanner (System.in);
    father () throws WrongAge {
  System.out.print("Enter Father's age :");
      age = in.nextInt();
      if ( age < 0 )
       throws new WrongAge (age);
    }
 }
```

```
class son extends father {
Scanner in = new Scanner (System.in);
    int fage;
  son (father f) throws WrongAge {
    this.fage = f.age ;
  System.out.print ("Enter son's age : ");
    this.age = in.nextInt();
      if (this.age <0)
     throw new WrongAge (age);
      if (this.age > f.age)
     throw new WrongAge (age);
  }
}
  class ages {
  public static void main (String args[]) {
    try {
       father f = new father ();
        son s = new son (f);
  }
   catch (Exception e) {
   System.out.println (e);
  }
 }
}
```

OUTPUT :-

```
Enter the father's age : 40
Entel the son's age : 56
enter correct age 56 is invalid.
```