7) Demonstrate generics with multiple object parameters.

A) import Java.util.Scanner;
class twoGen <T, V>
{ T ob1;
  V ob2;
    two Gen (T o1, V o2)
  { ob1 = o1;
    ob2 = o2; }
  void showTypes()
  System.out.println ("Type of T is " + ob1.getclass().
                          .getName());
  System.out.println("Type of V is" + ob2.getclass().getName());

T getob1() { return ob1; }
V getob2() { return ob2; }
}

class multiplegen
{
  public static void main (String args[])
  twoGen <Integer, String> tg = new twoGen <Integer, String>
                          (10, "hello world");

  tg.showTypes();
  int v = tg.getob1();
  System.out.println ("value of int :" + v);
  String str = tg.getob2();
  System.out.println("value of string :" + str);

twoGen < Boolean, Float> tg1 = new twoGen <Boolean, Float>
                          (true, 3.14 f);
tg1.showtypes();

boolean b = tg1.getob1();
System.out.println ("value of boolean :" + b);
float f = tg1.getob2();
System.out.println ("value of float :" + f);
}

8) Exceptions in inheritance tree. Father → Son derived
in Father, constructor takes age and throws exception
Wrong age() when input age < 0.
in Son, constructor throws exception if son age >= fage.

A) class WrongAge extends Exception
{
  private int a;
  WrongAge (int b)
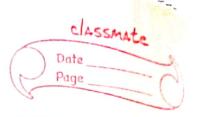 { a = b;
  }
  public String toString() { return "WrongAge (" + a ")";
}

class father
{
  int fatherage;
  father() throws WrongAge
  Scanner sc = new Scanner (System.in);
  System.out.println ("Enter father's age");
  fatherage = sc.nextInt();

```
        if (fatherage < 0)
              throw   new   WrongAge (fatherage);
    }  else System.out.printh ("correct father's age");
    }
}

class son extends father
{ int sonage;
  son ()  throws WrongAge
{
  Scanner sc = new  Scanner (System.In);
  System.out.printh ("enter son's age");
  sonage = sc.nextInt();
      if (sonage >= fatherage)
              throw  new  WrongAge (sonage);
    } else System.out.printh ("No error");
}

class exception-tree
{
  public static void main (string args[])
  { try { Son  s = new son();
    } catch (WrongAge  e)
        System.out.printh ("Caught "+e);
    }
  }
}
```