program-7:

1. Write a program to demonstrate gene
with multiple object parameters.

```
class Gen<T> {
T ob;
Gen(T
```

```
class TwoGen<T, V>
{
T ob1;
V ob2;
TwoGen(T o1, V o2)
{
ob1 = o1;
ob2 = o2;
}
void showTypes() {
System.out.println("Type of T is " + ob1.
    class().getName());
System.out.println("Type of V is " + ob2.ge
    class().getName());
}
T getob1() {
return ob1;
}
V getob2() {
return ob2;
}
}
class SimpGen {
public static void main(String args[])
{
```

```
Two Gen<Integer, String> tgobj = new Two
Gen<Integer, String>(88, "Generics");
tgobj.showTypes();
int v= tgobj.getob1();
System.out.println("value:"+v);
String str= tgobj.getob2();
System.out.println("value:"+str);
}
}
```

Execute | Share Source File STDIN

```
}

T getob1() {

return ob1;

}

V getob2() {

return ob2;

}

}


 public class SimpGen {

public static void main(String args[])
{

TwoGen<Integer, String> tgObj =

new TwoGen<Integer, String>(88,"Generics");

tgObj.showTypes();

int v = tgObj.getob1();

System.out.println("value: " + v);

String str = tgObj.getob2();

System.out.println("value: " + str);

}

}
```

Result

```
$javac SimpGen.java

$java -Xmx128M -Xms16M SimpGen
Type of T is java.lang.Integer
Type of V is java.lang.String
value: 88
value: Generics
```

program - 8

// write a program that demonstrates handling of exceptions in inheritance tree. create a base class called 'Father' and derived class called "son" which extends the base class. In father class, implement a constructor which takes the age & throws the exception wrong age(). when input age < 0. In son class, implement a constructor that cases both father & son's age & throws an exception if son's age is >= father's age

```
// class wrong age extends exception
{
  public String toString()
  {
    return "please enter the right age.";
  }
}
class father
{
int age;
Father(int age1)
  {
    age = age1;
    System.out.println("Father age:"+age);
}
class son extends father
{
  no(int age1);
```

```
    super (age);
    System.out.println ("son age :" +age);
    }
class final
public static void main(String args[]) throws
    wrongage
    {
    int i = args.length;
    int j = integer.parseInt (args[0]);
    int k = integer.parseInt (args[i]);
    if (i<0 || i<=k)
    {
    throw new wrongAge();
    }
    else
    {
    Father f = new Father(j);
    Son s = new Son (k);
    }
}
}
```