

write a program to demonstrate generics with multiple object parameters.

```
import java.util.*;  
class ThreeGen<T, V, W> {  
    T ob1;  
    V ob2;  
    W ob3;  
    ThreeGen(T o1, V o2, W o3) {  
        ob1 = o1;  
        ob2 = o2;  
        ob3 = o3;  
    }  
  
    void showTypes() {  
        System.out.println("Type of T is" + ob1.getClass().getName());  
        System.out.println("Type of V is" + ob2.getClass().getName());  
        System.out.println("Type of W is" + ob3.getClass().getName());  
    }  
  
    T getob1() {  
        return ob1;  
    }  
  
    V getob2() {  
        return ob2;  
    }  
  
    W getob3() {  
        return ob3;  
    }  
}  
  
class SimpleGen {  
    public static void main (String args[]) {  
        Scanner sc = new Scanner (System.in);  
        int a;
```

```

String b;
float c, c1;
char a1;
double d1;
System.out.println("Enter a integer, string, float");
a = sc.nextInt();
b = sc.next();
c = sc.nextFloat();
Threegen < Integer, String, float > tgObj =
new Threegen < Integer, String, float > (a, b, c);
tgObj.showTypes();
int v = tgObj.getObj1();
System.out.println("value 1: " + v);
String str = tgObj.getObj2();
System.out.println("value 2: " + str);
float x = tgObj.getObj3();
System.out.println("value 3: " + x);
System.out.println("Enter a character, double, float");
a1 = sc.next().charAt(0);
d1 = sc.nextDouble();
c1 = sc.nextFloat();
Threegen < character, double, float > tgObj1 =
new Threegen < character, double, float > (a1, d1, c1);
tgObj1.showTypes();
char v1 = tgObj1.getObj1();
System.out.println("value 1: " + v1);
double str1 = tgObj1.getObj2();
System.out.println("value 2: " + str1);
float x1 = tgObj1.getObj3();
System.out.println("value 3: " + x1);
}

```