

LAB 7:

```
class TwoGen<T, V> {

    T ob1;

    V ob2;

    // Pass the constructor a reference to
    // an object of type T and an object of type V.

    TwoGen(T o1, V o2) {

        ob1 = o1;

        ob2 = o2;

    }

    // Show types of T and V.

    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;

    }

}

// Demonstrate TwoGen.

class generics {

    public static void main(String args[]) {

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

        // Show the types.

        tgObj.showTypes();

    }

}
```

```
// Obtain and show values.
```

```
int v = tgObj.getob1();
```

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

```
String str = tgObj.getob2();
```

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

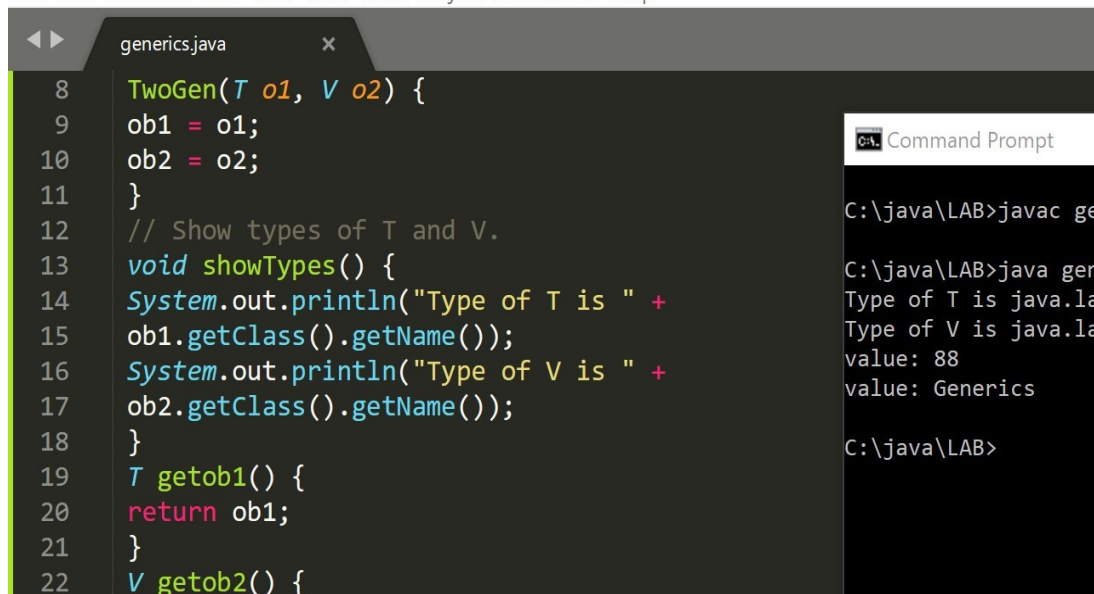
```
}
```

```
}
```

OUTPUT:

C:\java\LAB\generics.java (covid 19) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help



The screenshot shows the Sublime Text editor with a file named 'generics.java' open. The code in the editor is as follows:

```
8  TwoGen(T o1, V o2) {
9  ob1 = o1;
10 ob2 = o2;
11 }
12 // Show types of T and V.
13 void showTypes() {
14 System.out.println("Type of T is " +
15 ob1.getClass().getName());
16 System.out.println("Type of V is " +
17 ob2.getClass().getName());
18 }
19 T getob1() {
20 return ob1;
21 }
22 V getob2() {
```

Overlaid on the right side of the editor is a Windows Command Prompt window. It shows the following commands and output:

```
C:\java\LAB>javac ge
C:\java\LAB>java gen
Type of T is java.la
Type of V is java.la
value: 88
value: Generics
C:\java\LAB>
```