

LAB 7

```
Project Name: generic
1 ~ class Gen<T> {
2
3   T ob; // declare an object of type T
4
5   // Pass the constructor a reference to
6
7   // an object of type T.
8
9   Gen(T o) {
10
11     ob = o;
12
13   }
14
15   // Return ob.
16
17   T getob() {
18
19     return ob;
20
21   }
22
23   // Show type of T.
24
25   void showType() {
26
27     System.out.println("Type of T is " +
28
29       ob.getClass().getName());
30
31   }
32
33 }
34
35 ~ public class GenDemo {
36
37   public static void main(String args[]) {
38
39     Gen<Integer>iob;
40
41     iob = new Gen<Integer>(88);
42
43     iob.showType();
44
45     // Get the value in iob. Notice that
46
47     // no cast is needed.
48
49     int v = iob.getob();
50
51     System.out.println("value: " + v);
52
53     System.out.println();
54
55     // Create a Gen object for Strings.
56
57     Gen<String>strob = new
58     Gen<String>("Generics Test");
59
60     // Show the type of data used by strOb.
61
62     strob.showType();
63
64     // Get the value of strOb. Again, notice
65
66     // that no cast is needed.
67
68     String str = strob.getob();
69
70     System.out.println("value: " + str);
71
72   }
73
74 }
75
76
```

Execute Mode, Version, Inputs & Arguments

JDK 11.0.4



Interactive

CommandLine Arguments

Execute



Result

compiled and executed in 0.984 sec(s)

Type of T is java.lang.Integer
value: 88

Type of T is java.lang.String
value: Generics Test