

```
public class Gen <T>{
    T ob; // declare an object of type T
    // Pass the constructor a reference to
    // an object of type T.
    Gen(T o) {
        ob = o;
    }
    // Return ob.
    T getob() {
        return ob;
    }
    // Show type of T.
    void showType() {
        System.out.println("Type of T is " +
            ob.getClass().getName());
    }
}
```

```
public class GenericsDemo {

    public static void main(String[] args) {
        // Create a Gen reference for Integers.
        Gen<Integer> iOb;
        // Create a Gen<Integer> object and assign its
        // reference to iOb. Notice the use of autoboxing
        // to encapsulate the value 88 within an Integer object.
        iOb = new Gen<Integer>(88);
        // Show the type of data used by iOb.
        iOb.showType();
        // Get the value in iOb. Notice that
        // no cast is needed.
        int v = iOb.getob();
        System.out.println("value: " + v);
        System.out.println();
        // Create a Gen object for Strings.
        Gen<String> strOb = new Gen<String>("Generics Test");
        // Show the type of data used by strOb.
        strOb.showType();
        // Get the value of strOb. Again, notice
        // that no cast is needed.
        String str = strOb.getob();
        System.out.println("value: " + str);
    }
}
```