**Generics**

At its core, the term generics means parameterized types. Parameterized types are important because they enable you to create classes, interfaces, and methods in which the type of data upon which they operate is specified as a parameter.

A class, interface, or method that operates on a parameterized type is called generic, as in generic class or generic method.

Ex>

**class Gen<T>{**

**T ob**

**Int x ;**

**Gen(T o) {**

**ob = o;**

**}**

**T getob() {**

**return ob;**

**}**

**void showType() {**

**System.out.println("Type of T is " + ob.getClass().getName());**

**}**

**}**

class GenDemo {

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.

**Gen<Float> iOb = new Gen<Float>(88.5);**

**Gen g =new Gen(10);**

**Gen<Integer> iob = new Gen<Integer>(10);**

**Gen<Double> iob = new Gen<Double>(10.2345);**

// 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);

}

}

**Advantages:**

[**https://www.surveymonkey.com/r/YTCJLQH**](https://www.surveymonkey.com/r/YTCJLQH)

**1. Code Reusability**

**2. Rapid Application Development**

**3. Type Safe**

**4. No type casting required.**

**General Notations used in Generics:**

**--------------------------------------------**

**T --> Type**

**e --> Element.**

**K --> KeyType**

**V --> Value Type**