Unit 5
Generic types

UD

Sep 2023

Introduction on Generics

Casting needed – without Generics:

```
class Gen {
        private Object obj; // Take up any
values
        Gen(Object obj) { this.obj = obj; }
        Object getObj() { return obj;}
}

class GenDemo {
   public static void main(String[] args) {
        Gen o1 = new Gen(50); // int
        int i = (int) o1.getObj(); // Casting
        System.out.println("value: " + i);


        Gen o2 = new Gen("TMSL"); //
String
        String str = (String) o2.getObj(); //
C..
        System.out.println("value: " + str);

   }
}
```

Casting not needed - with Generics:

```
class Gen<T> { // Generic with Diamond operator
   T obj;     // obj is of type T
   Gen(T obj) { this.obj = obj; }
   T getObj() {return obj;}
    void showType() { System.out.println("Type:  " +
obj.getClass().getName());  }
}

class GenDemo {
   public static void main(String[] args) {
       Gen<Integer> o1=new Gen<Integer>(50);
        o1.showType();
        int i = o1.getObj();// No casting needed
        System.out.println("value: " + i);

        Gen<String> o2 = new Gen<String>("TMSL");
        o2.showType();
        String str = o2.getObj();// No casting needed
        System.out.println("value: " + str);

    }
}
```

Why Generics?

1. Stronger type checks at compile time

- Early detection of error

2. Elimination of casts

- Casting required without generics:

```
List list = new ArrayList();
list.add("hello");
String s = (String) list.get(0);
```

- Casting not required with generics:

```
List<String> list = new ArrayList<String>();
list.add("hello");
String s = list.get(0);   // no cast
```

3. Implement generic algorithms

- Generics used for implementing generic algorithms
  - . Refer demo on previous slide
- Works on collections of different types
- Customizable, type safe and easier to read

4. Generic code: Reusable feature in Java. Similar features in **C#**, C++ (templates)

Demonstration on GenSimpleStack

Please click here for demo programs.

Compile using:

javac GenSimpleStackDemo.java IGenSimpleStack.java GenSimpleStack.java SimpleStackEsc.java

Run using:

java GenSimpleStackDemo.java

Output:

```
Demonstrating String Stack
Pushing: alpha beta gamma
Popping: gamma beta alpha

Demonstrating Integer Stack
Pushing: 10 20 30
Popping: 30 20 10
```