

```
class Box<T> {  
    private T t;  
  
    public void add(T t) {  
        this.t = t;  
    }  
  
    public T get() {  
        return t;  
    }  
}
```

```
public class GenericsTester {  
    public static void main(String[] args) {  
        //type inference  
        Box<Integer> integerBox = new Box<>();  
        //unchecked conversion warning  
        Box<String> stringBox = new Box<String>();  
  
        integerBox.add(new Integer(10));  
        stringBox.add(new String("Hello World"));  
  
        System.out.printf("Integer Value : %d\n",  
integerBox.get());  
    }  
}
```

```
System.out.printf("String Value :%s \n",  
stringBox.get());  
}  
}
```

T->Type

E->Element

K->Key

N->Number

V-Value

class genericmethod

```
{  
public static <E> void printarray(E[] elements)  
{  
for(E e:elements)  
{  
System.out.println(e);  
}  
System.out.println();  
}  
public static void main(String[]args)  
{  
Integer[] intarr={10,20,30,40};  
Character[] charr={'A','M','I','T'};
```

```
System.out.println("Printing Integer Array");
```

```
printarray(intarr);
```

```
System.out.println("Printing String Array");
```

```
printarray(charr);
```

```
}
```

```
}
```