



SNS COLLEGE OF TECHNOLOGY

(AN AUTONOMOUS INSTITUTION)

COIMBATORE – 35

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



UNIT 4

What is Generics?

- The term GENERICS means parametrized types.
- Generics are a way to create reusable code that work with different types.
- It Maximize code reusability.
- They allow for the creation of classes, interfaces and methods that can be parametrized with different types.
- It is used for Type safety performance and cleaner code.
- Generics also allow for easier maintenance and fewer bugs.

Advantages of Java Generics

1) Type safety:

```
List list = new ArrayList();
```

```
list.add(10);
```

```
list.add("10");
```

With Generics, it is required to specify the type of object we need to store.

```
List<Integer> list = new ArrayList<Integer>();
```

```
list.add(10);
```

```
list.add("10");// compile-time error
```

2) Type casting is not required:

```
List list = new ArrayList();
```

```
list.add("hello");
```

```
String s = (String) list.get(0);//typecasting
```

After Generics, we don't need to typecast the object.

```
List<String> list = new ArrayList<String>();
```

```
list.add("hello");
```

```
String s = list.get(0);
```

3) Compile-Time checking:

```
List<String> list = new ArrayList<String>();
```

```
list.add("hello");
```

```
list.add(32);//Compile Time Error
```

Syntax to use generic collection;

```
ClassOrInterface<Type>
```

Example to use Generics in java;

```
ArrayList<String>
```

Type parameters;

T-Type

E-Element

K-key

N-Number

V-Value

Generic class

```
class MyGen<T>{  
    T obj;  
    void add(T obj){ this.obj=obj;}  
    T get(){return obj;}  
}
```

Program using Generic class

```
class TestGenerics3{  
    public static void main(String args[]){  
        MyGen<Integer> m=new MyGen<Integer>();  
        m.add(2);  
        //m.add("vivek");//Compile time error  
        System.out.println(m.get());  
    }  
}
```

Output:

2

Example for Generics in JAVA

```
1. import java.util.*;
2. class TestGenerics1 {
3.     public static void main(String args[]){
4.         ArrayList<String> list=new ArrayList<String>();
5.         list.add("rahul");
6.         list.add("jai");
7.         //list.add(32);//compile time error
8.
9.         String s=list.get(1);//type casting is not required
10.        System.out.println("element is: "+s);
11.
12.        Iterator<String> itr=list.iterator();
13.        while(itr.hasNext()){
14.            System.out.println(itr.next());
15.        }
16.    }
17. }
```

Output:

```
element is: jai
rahul
jai
```