

Hashtable

```
static Hashtable hashTableFunc(Hashtable h)
{
    h.put("four", new Integer(4));
    return h;
}
```

In Main :

```
Hashtable numbers = new Hashtable();
numbers.put("one", new Integer(1));
numbers.put("two", new Integer(2));
numbers.put("three", new Integer(3));
```

```
numbers=hashTableFunc(numbers);
```

After executing the code above, aspectj produces the following inner state :

```
hashTableFunc{two=2, one=1, three=3}{two=2, one=1, three=3, four=4}
```

```
### : executed functions name
```

```
### : input
```

```
### : output
```

ArrayList

```
static ArrayList arrayListFunc(ArrayList aList)
{
    aList.add(100);
    return aList;
}
```

In Main :

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

```
list.add("e1");
list.add("e2");
```

```
int x =0;
a=arrayListFunc(a);
```

After executing the code above, aspectj produces the following inner state :

```
arrayListFunc[e1, e2][e1, e2, 100]
```

```
### : executed functions name
```

```
### : input
```

```
### : output
```

Note :Array type collection objects like Stacks,Queues have the same inner state correspondence with ArrayLists when passed and/or returned.