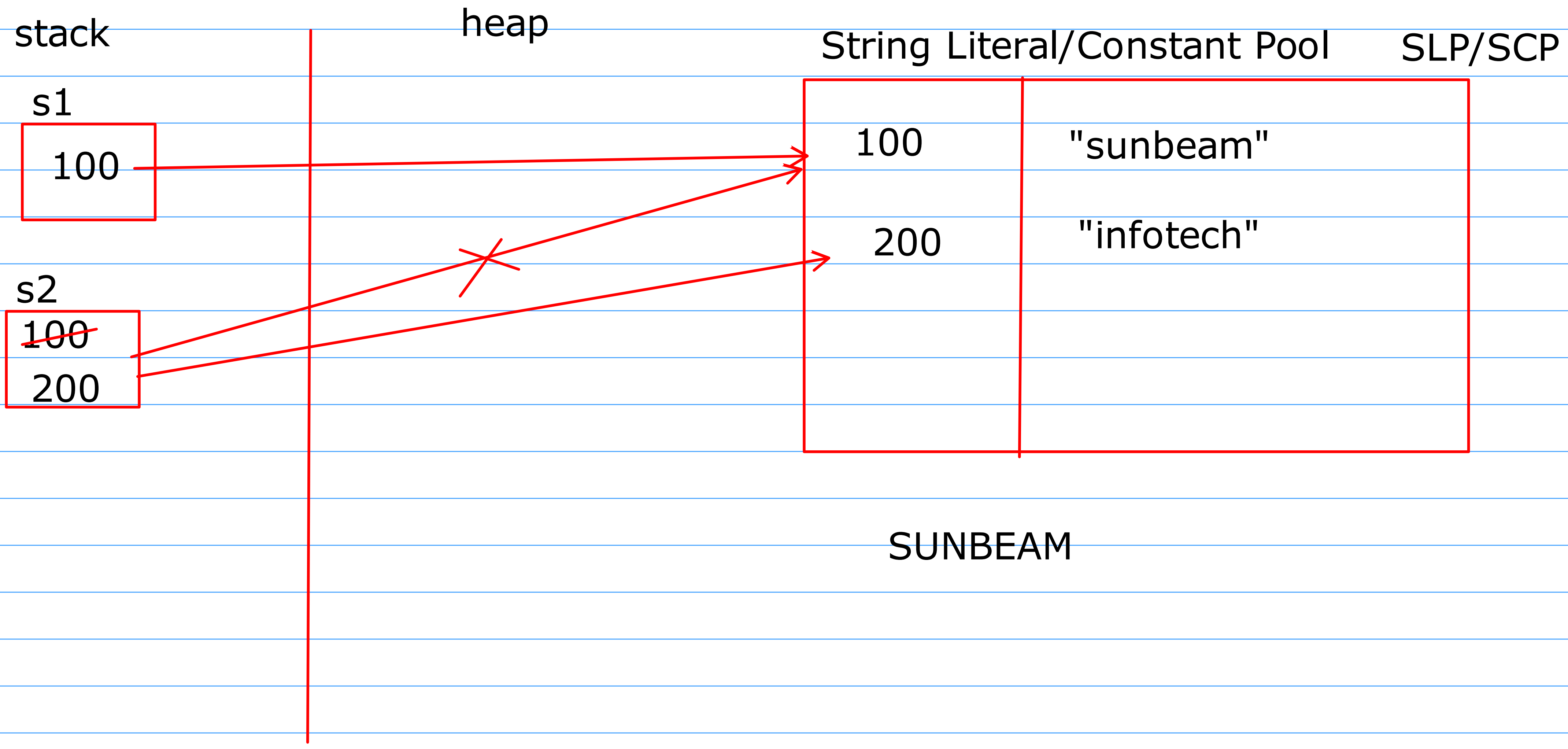
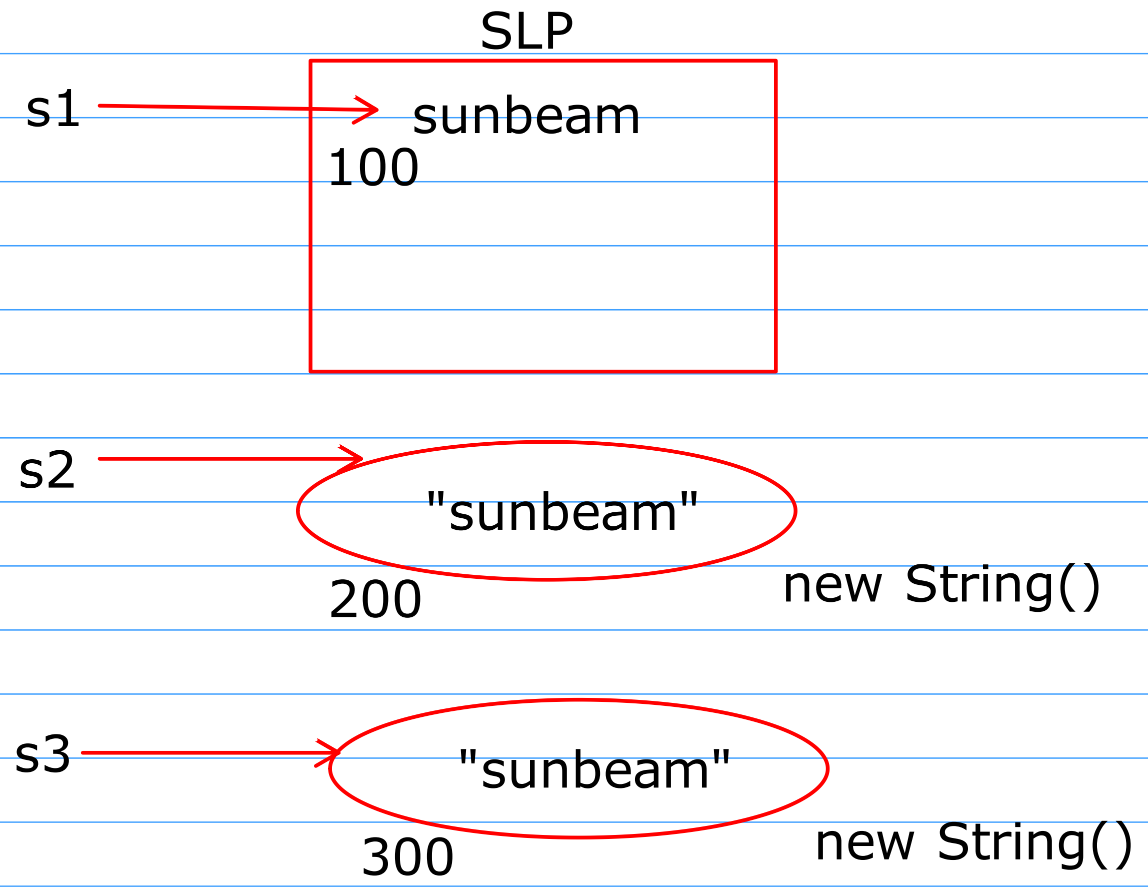
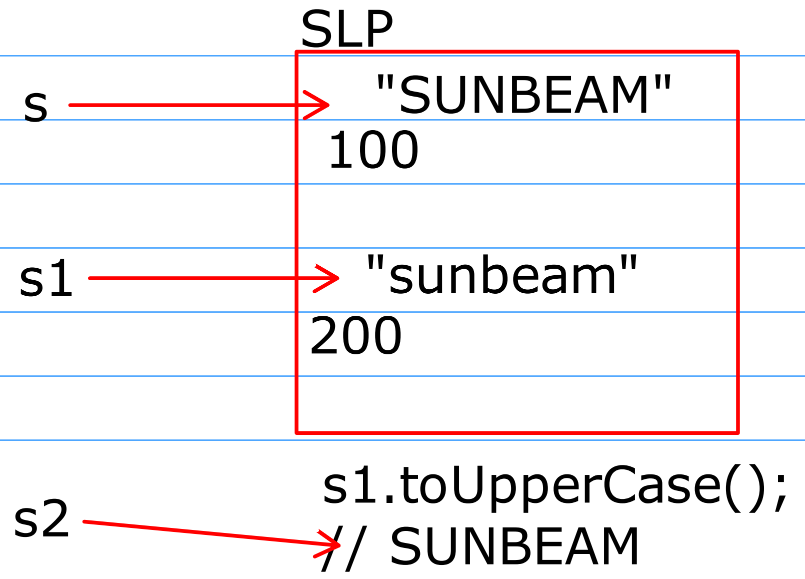
String - Sequence of characters

```
class String{                    String name = "Anil";// String Literal
                                 ref = address of an object
}                                ref = null
```





String in java is immutable



```
class VoterInformation{
String voterId;
String name;
String city;
String pin;
String mobile;
String Nationality;
}
```

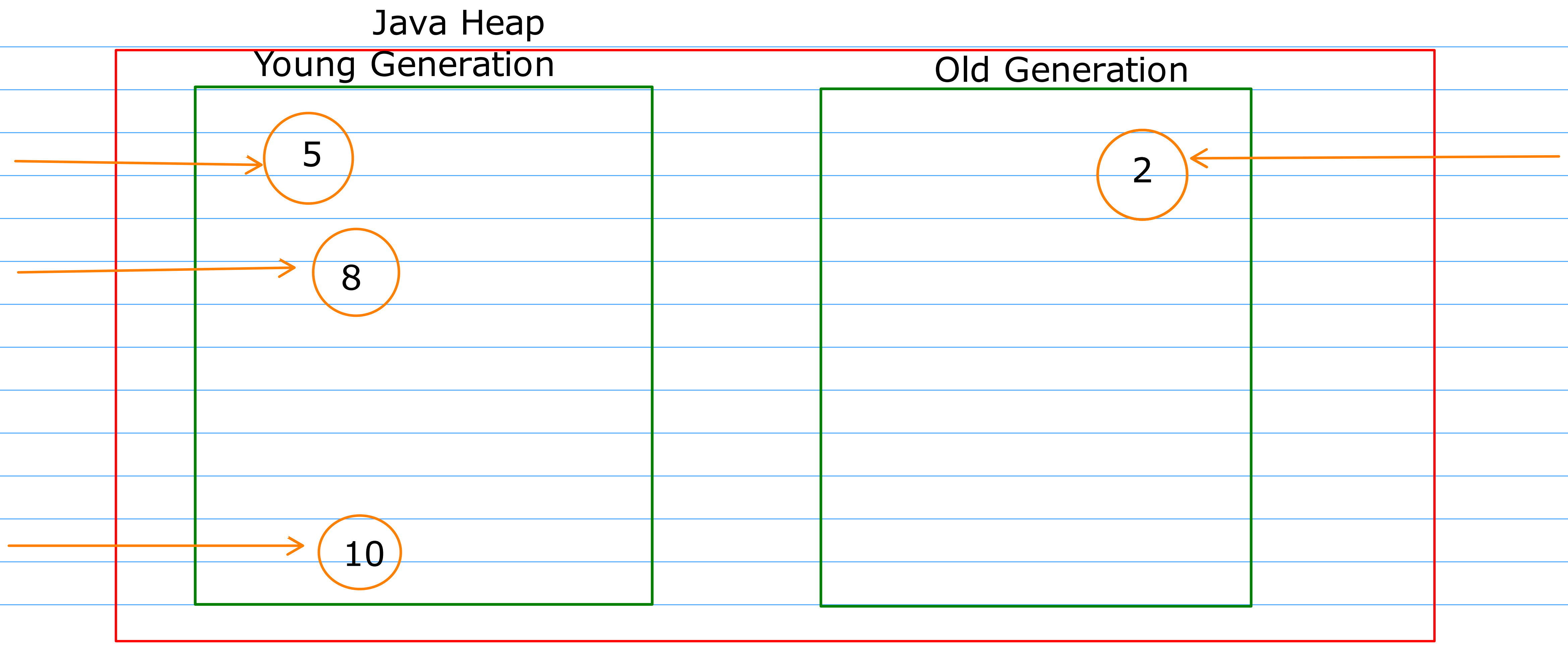
v1	v2	v3
v01	v02	v03
Anil	Mukesh	Anil
Pune	Pune	Pune
411057	411057	411057
9876543210	8976543210	7896543210
100 bytes	100 bytes	100 bytes

100000 * 100 = 1000,00,00

100000*60 = 60,00000 + 40 bytes

100000 * 40
50000 * 20
40 bytes

Garabge Collector



```
Date d1 = new Date();
Date d2 = d1
```

case -1

```
Date d1 = new Date(); // GC
d1 = null;
```

case-2

```
Date d1 = new Date();//GC
d1 = new Date();
```

```
void method(){
Date d1 = new Date(); // GC  method();
}
```

Mark and Compact Algorithm

1. Minor GC
2. Major GC

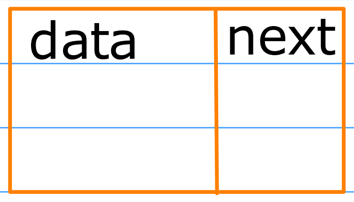
String

-

Generics

Collection Framework

LinkedList



Node

1.5 -> Genrics -> Type Safety

upper Bound -> Number

Number

Lower Bound -> Integer

Employee

Lower Bound - Manager Salesman

# Generic
1. Class
2. Method
3. Interface

Generic Class Reference
Box<? > b;

Generic Method;

void display(Box <>b)

<T>void displayArray(T []arr)

1. Comparable
2. Comparator

```
interface Comparable<T>{
        int compareTo(T o);
}
```

Student

compareTo(Date d);this->d1, o->d2

```
equals(Object obj); // this->d1 , obj=d2
{
Downcasting
}
```

Date d1;
Date d2;

d1.equals(d2);