→ Context after todays Session ↔ After todays Context

// class Main() {

- fun1()
- fun2()
- fun3()
- fun4()
- fun5()

→ very long
→ Readibility
→ Debugging code becom very very difficult
→ update
→ Testing

→ modular / functional

psvm() {

}

}

// Object orient Programming / Java

Class — blue print

Object — Real creation of Blu Print
Real instance of class

Class Car → Blueprint

attributes of a class / data:
- colour
- seating capacity
- airbags
- mplage
- horse power

functionals of a class:
- drive()
- break()
- music()
- gears()
- A/C()

Object: vartha's Car

colour = "Red"
seating capacity = "4"
airbags = "4"
mplage = "25"
horse power = "1200cc"
drive()
break()
music()
gears()
A/C()

Object: Shlok Car

colour = Black
seating capacity = 7
airbags = 1
mplage = 20
horse power = 1200
drive()
break()
music()
gears()
A/C()

Syntax: (small letter)

```
class rect
{
    int l;
    int b;
    int area() {
        return l*b;
    }
    int permeter() {
        return 2(l+b)
    }
}
// Blue print of rectangle
```

Syntax: To create objects

```
classname objname = new classname();

rect obj = new rect();
```

objcreat

↳ how we can create an Object

```
obj.l = 4; obj.b = 6;
```

obj { l:4, b:6 }

```
int n = obj.area()
print(n)
```

→ Calling functional

↳ obj.area()

{l:4, b:6} → {return l*b; → 24}

// Diff Obj & Obj reference ?

```
class rect
    int l;
    int b;
    int area( ) {
        return l * b;
    }
    int permeter( ) {
        return 2(l + b)
    }
}
```

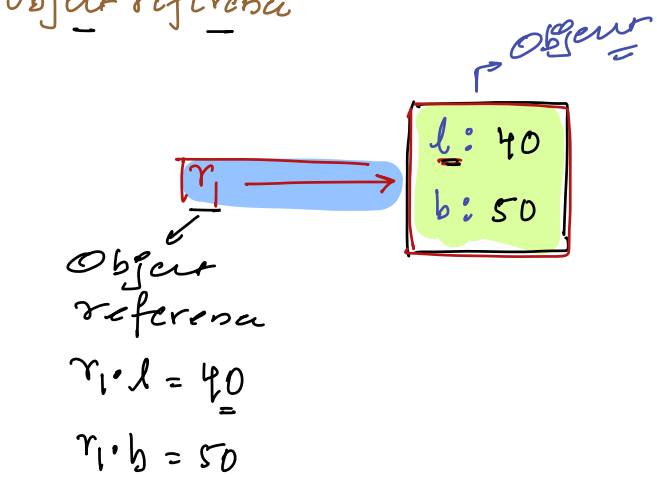
rect r_1; } object not created

r_1 = new rect();

↳ creation of Object

↓

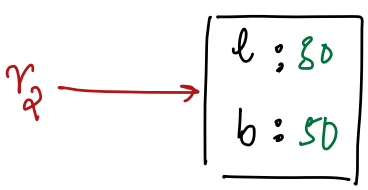
Object reference



$r_1 \cdot l = 40$

$r_1 \cdot b = 50$

rect r_2 = new rect()

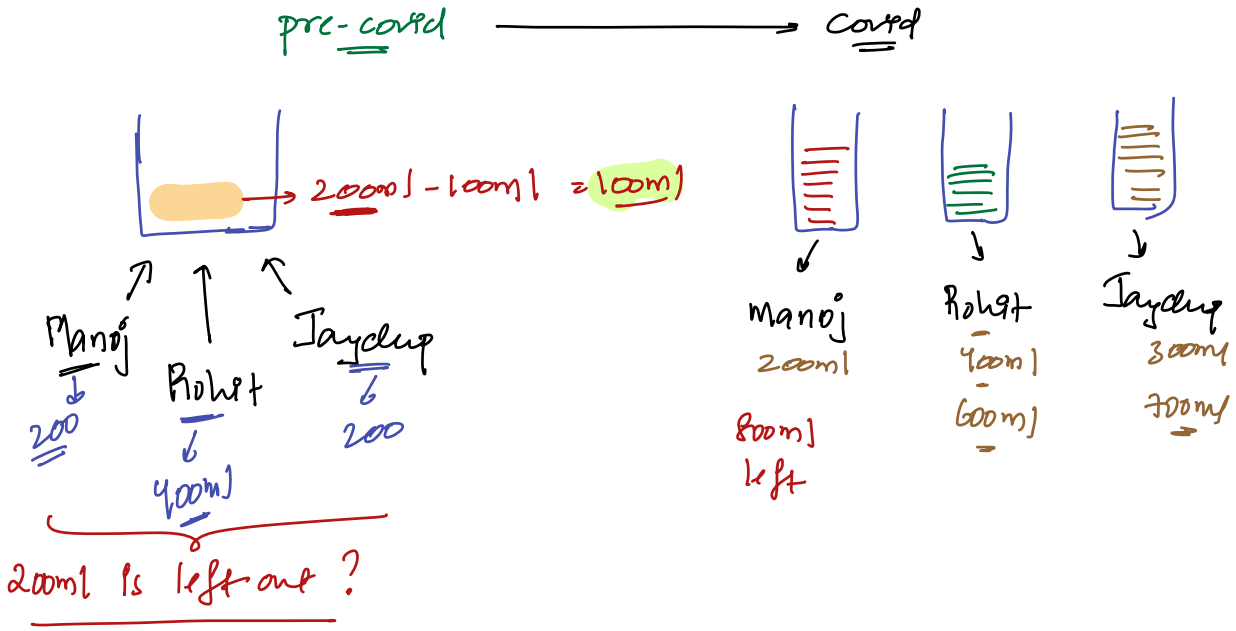


$r_2 \cdot l = 30$

$r_2 \cdot b = 50$

// what's the use of object reference?

// Story:



All 3 are using same bottle

1) Change done by a person it's also effection other person

// All 3 are having their own bottles

1) Change in 1 bottle won't effect other bottles

2) Object refences: Multiple object reference can point to same Object

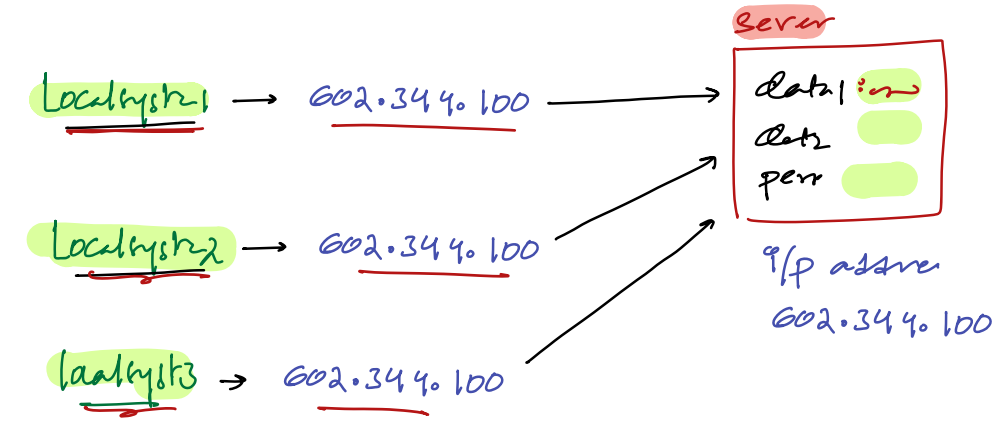
```
Rect a1 = new Rect()
a1.l = 30, a1.b = 40
```

a1 → l = ~~30~~ 50
a2 → b = 40

```
Rect a2 = a1
```

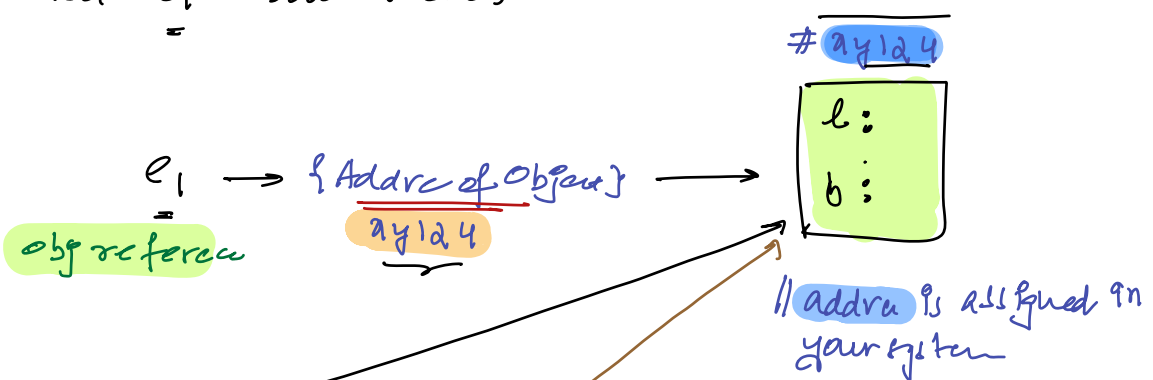
} a2 will now point to same object a1 is point

```
a2.l = 50
print(a1.l) => {=>50}
```



// Local system → { g/p address } → { We are going there getting data }

// Rect e_1 = new Rect()



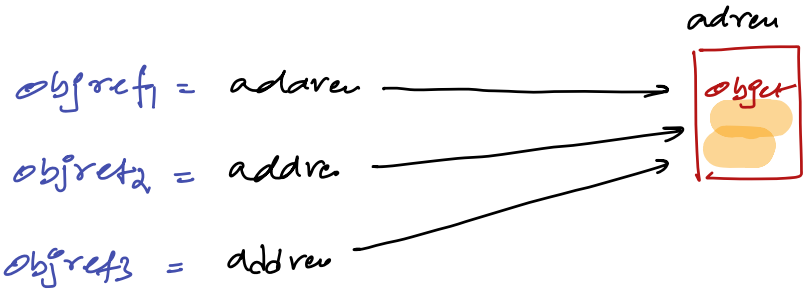
Rect e_2 = e_1 ;

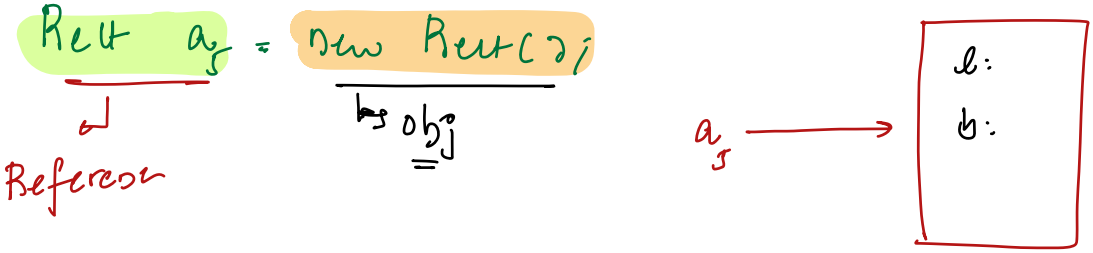
#ay124 ← #ay124

Rect e_4 = e_2 ;

#ay124 = #ay124

// In our obj referenc we just store address of an object & using that we are able to acw the data of that object present at that address





Rect a; → {object not created}

↳ object Reference is created