Class → Blueprint /
Skeleton /
Template

short interesting

Object → Instance of the class

Eg
```
class Employee {
    public string name
    public int age
    public int id
    public long mob_no
    public string department
}
```

How to create an Employee.?

```
Employee x = new Employee()
```

data type

```
x.name = "Aman"
x.age = 25
```

Constructor

1) Function
2) Same name as class
3) No return type

- Constructor is called when object is created

```
class Student {
    String name
    int age
    int id

    Student(String name, int age, int id) {
        this.name = name
        this.id = id
        this.age = age
    }
}
```

```
Student s1 = new Student (
    "Ayush", 17, 100345678)

print s1.id= 100345678
```

Q Create a class rectangle
Length, Breadth

Functions: 1) Find area of rect
2) Find if rect is a square or not

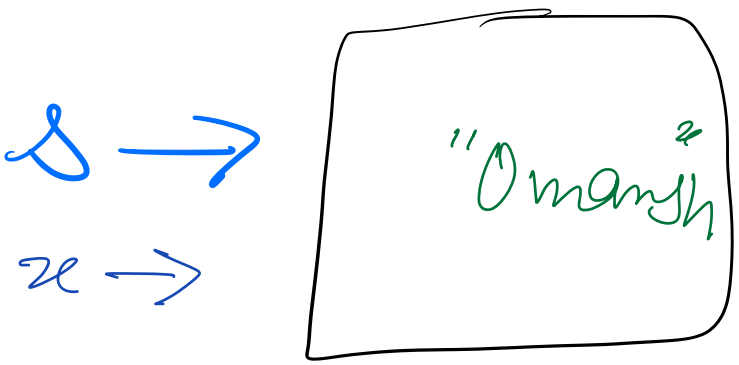
```
class Rectangle {
    int length
    int breadth
    Rectangle (int l, int b){
        this.length = l
        this.breadth = b
    }
    int area ( ) {
        return length * breadth
    }
    boolean isSquare {
        return length == breadth
    }
}
```

Shallow & Deep Copy

Student s = new Student ("Aman");
Student x

x = s

shallow copy



s.name = "Omansh"

x.name

s ⇒ Aman Add

x ⇒ "Aman Add

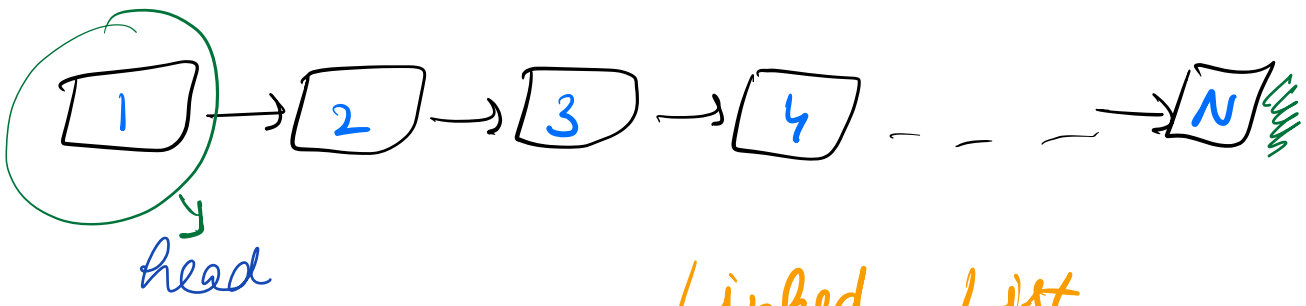
```
Student s = new Student ("Aman");
Student x = new Student ();
x.name = s.name
x.age = s.age
```

Deep Copy

Object Reference inside a class

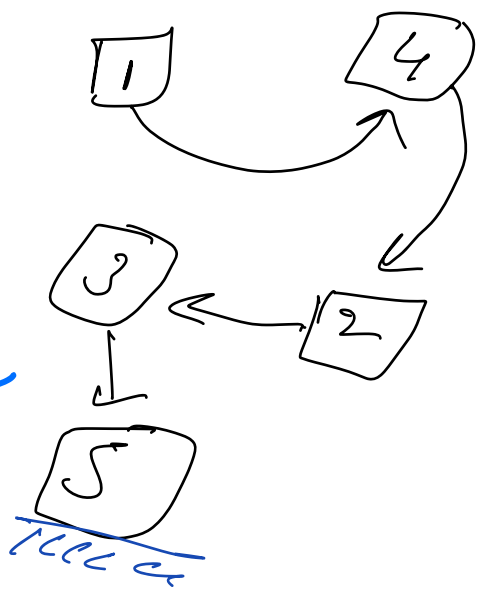
N objects of a class

List < Class > list



Linked List

```
class Node {
    int data
    Node next
    Node (int x) {
        this.data = x
        this.next = NULL
    }
}
```

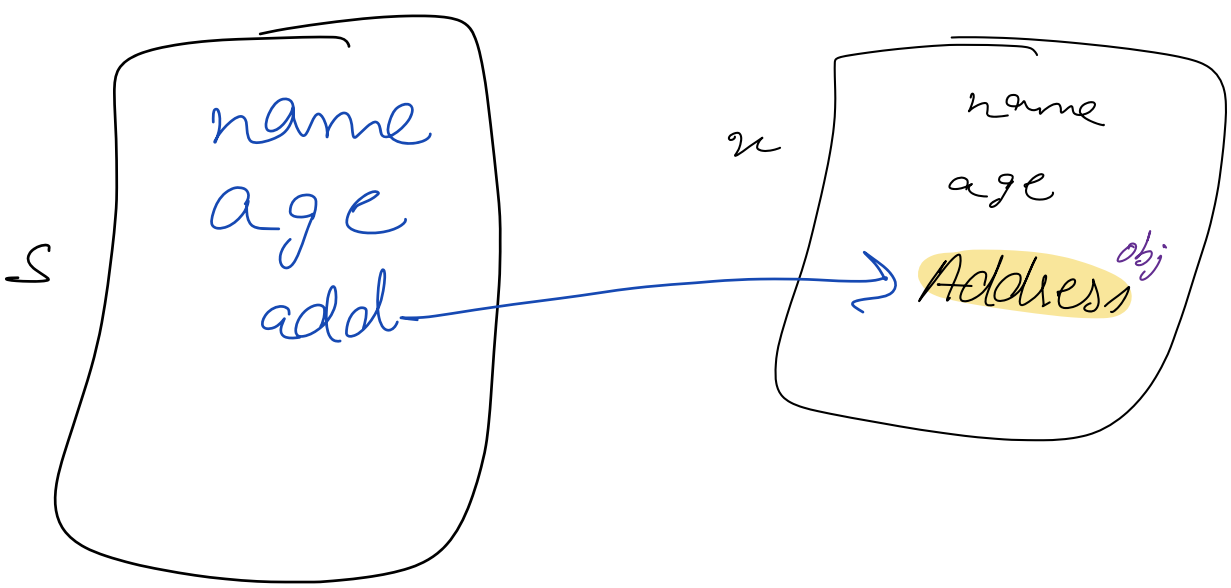


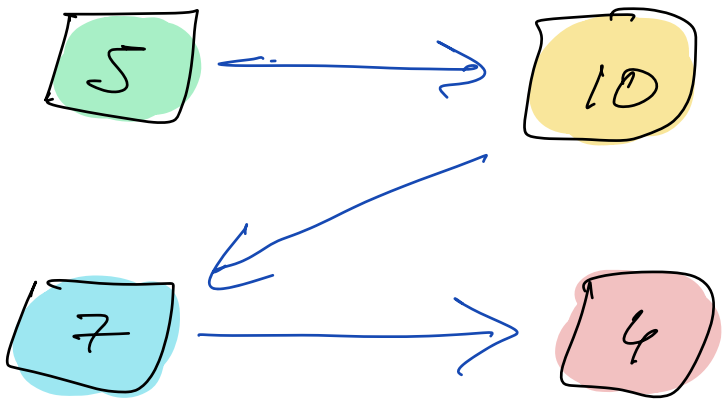
- Create rectangle of l=5 b=6
  Find the area (using class function)

```
Rectangle r = new Rectangle(5, 6)
print(r.area())
```

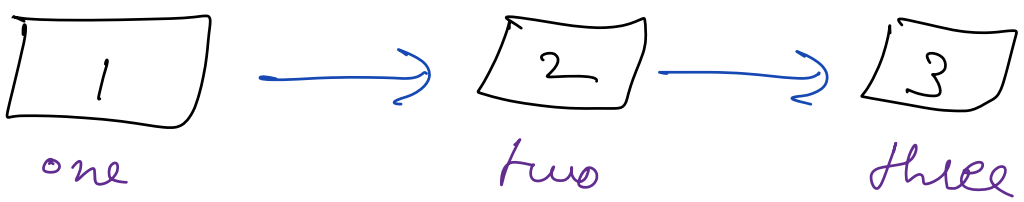
IsSquare

```
if (r.IsSquare())
    print(r.area())
```






```
Node one = new Node(1)
Node two = new Node(2)
Node three = new Node(3)
```




```
one.next = two
two.next = three
```

```
one.data    1
```

one.next.data 2
one.next.next.data 3
one.next.next.next
= NULL

Clone

```
for (i=0 ; i<100; i++)
    x = x.next
```

hs( )

```
for( Integer x : hs)
```