Today's Content

(a) class & object concept. — [LLD|HLD] [8am]

(b) Object & Object Reference.

(c) Object Reference as a member.

Link to Code Snippet: interviewbit.com/snippet/9ef73417e3ce565cd898/

Class: it is a blueprint.

 Blueprint (design)

Object: Real instance of the blueprint.
creation

 ⇒ converted blueprint into a house.

Class:
→ ① Attributes : Information
→ ② Functionality : Action

```
class Car {
    brand
    color
    Seater
    engine type

    break()
    Speed()
    music()
}
```

Mohit's Car
brand : tata
Color : black
Seater : 6
engine type : petrol

Attributes will change object to object

Ram's Car
brand : Audi
Color : white
Seater : 4
engine type : petrol

break() ←————————→ break()
Speed() ←————————→ Speed()
music() ←————————→ music()

```
class Student {
    string name
    int roll no
    int m1, m2, m3
    int totalmarks() {
        return m1+m2+m3
    }
    int maxMarks() {
        return Math.max(m1, Math.max(m2,m3))
    }
    void printname() {
        System.out.println(name)
    }
}
```

classname objName = new ClassName();

Student s = new Student();
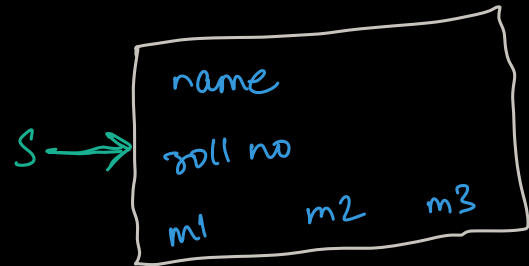
'.' (dot) operator

s.name = "sapna"
s.rollno = 45
s.m1 = 50
s.m2 = 60
s.m3 = 70

Object & Object Reference

```
class Student {
    string name
    int roll no
    int m1, m2, m3
    int total marks () {
        return m1+m2+m3
    }
    int max marks () {
        return Math. max (m1, Math. max(m2,m3))
    }
    void printname () {
        System. out. println (name)
    }
}
```
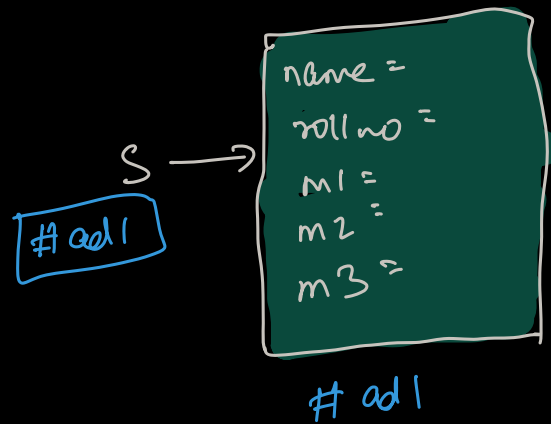
Student s1; → No errors

↳ object reference of class Student.

Student s = new Student();
↓
this line creates an object

name =
roll no =
m1 =
m2 =
m3 =

s ⟶

#ad1

ad1

Student s2 = new Student();

name =
roll no =
m1 =
m2 =
m3 =

s2 ⟶

#ad2

ad2
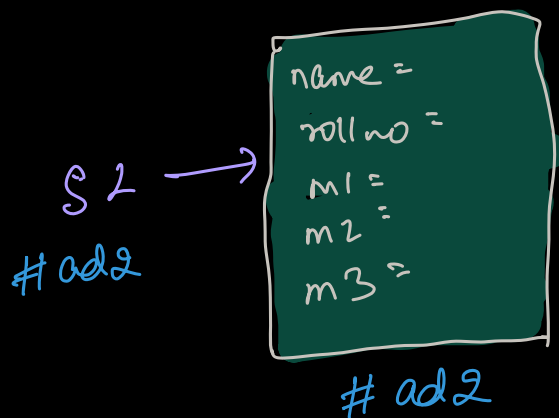
Object Reference can hold the address of that particular class object.

Student s3 = null; // No error

s3.m1 = 50 → null pointer exception! ERROR

Multiple Object Reference

```
class Student {

    string name
    int roll no
    int m1, m2, m3
    int total marks () {
    |   return m1+m2+m3
    }

    int max Marks () {
    |   return Math. max (m1, Math. max(m2, m3))
    }
    void printname () {
    |   System. out. println (name)
    }
}
```

Student s1 = new Student ()

s1 ────────→ ┌──────────┐
              │ name   │
#ad5   │ rollno   │
              │ m1 m2m3 │
    s2   ╱──→└──────────┘
    #ad5   #ad5   ↖
                              S3
Student s2 = s1;   #ad5
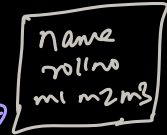
Student s3 = s2;
        #ad5   #ad5

Break till 8.40am

```
class pair {
|  int x, y
}
```

pair p1 = new pair();
                    // Creates an obj

p1
#ad7 → | x = 10
         | y = 20
              #ad 7

p1.x = 10
p1.y = 20
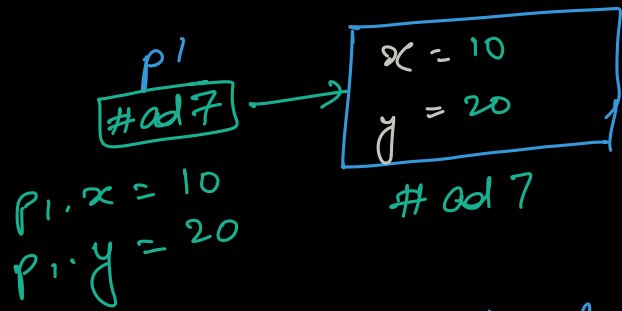
Constructor : Used to initialize the attributes of the class at the time of object creation.

```
class   pair {
|  int x.y;

      pair (int a, int b) {
      |    x = a
      |    y = b
      3
3
```

⊛ Name is same as class name

⊛ Similar to a function
       but NO return type.

pair p2 = new pair (20, 40)

P2
#ad8
          | x = 20
          | y = 40

      # ad 8