

# ⇒ Structure of an Interview

widely divided (100%)

PBC

SBC

DSA :-            60-70%            40-45%

(problem solving)

OOPS :-            20%            20%

CN :-            5-10%            X

DBMS :-            10-15%            10-15%

Dev :-            30-40%            40-50%

→ OOP's (Object oriented programming)

↳ Java class :-

↳ a class is a blueprint or template  
which is used to create objects.

→ Object :- Object can be anything

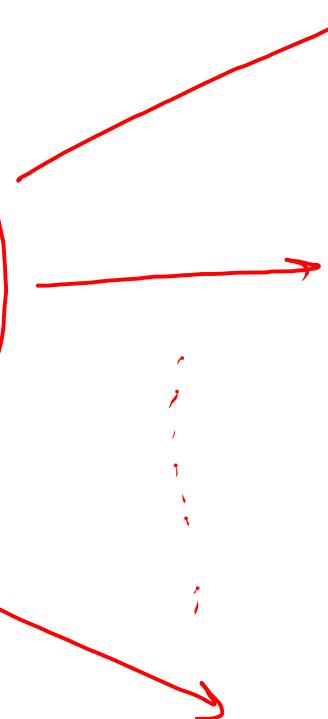
# Properties of a class :-

- ↳ A java class serves as a blueprint for creating objects and class doesn't take any memory.
- ↳ It can contain variables, methods, constructors and nested class, interfaces, ~~constructor~~.  
not in java
- Note :- Java language doesn't need destructor because we have garbage collector.
- ↳ class defines the behaviour of our objects.

~~Imagine~~

class c1

variables  
function / methods  
constructor  
interfaces  
etc.



obj 1

obj 2

obj n

# ⇒ Components of a java class

## 1) Class declaration :-

```
public class MyClass {  
    [ ]  
    }  
    Keyword
```

## 2) fields / instance variables :-

```
private int age;
```

3) Constructor :- constructor initialize objects of the class. They always have same name as of class

Gmp:- They are always called when an object is created

Syntax

```
public MyClass ( int age ) {  
    [ ]  
    }  
    this. age = age ;  
    optional  
    constructors doesn't have  
Gmp: any return type
```

Note :- Imp keywords :- this, new, static, final

4) Methods :- methods defines the behaviour of our object

```
public void display () {  
    System.out.println (age);  
}
```

5) Access modifier :- these are used to control the accessibility of class member ( fields, constructor, method )

syntax of  
a class

```
public class My Class {  
    → private int age; ✓
```

```
    public My Class (int age) {  
        this . age = age; ✓
```

```
        }  
        public void display () {
```

```
            System.out.println ("Methods run"); ✓
```

```
        }
```

```
}
```

## ⇒ 9mp properties of a class

1) Single responsibility principle :-

a class should have a single purpose and should be well defined

2) Open closed responsibility :- ( inheritance )

a class should be able to extend another class without modifying class.

3) Liskov substitution responsibility :- a derived class / child class should be substitutable for its base class / parent class -

4) Dependency inversion principle :- (Abstraction)

High level modules should not import anything from a low level module & should depend on abstract.

5) Interface Segregation Principle :- (Abstraction)

A client should not be exposed to methods it does not need.

⇒ Basic rules of for creating class :-

- 1) The "class" Keyword should be used
- 2) The name of class should always with upper case letter
- 3) A java file can contain any no. of class.
- 4) one class should only inherit another class properties

⇒ Type of class

- 1) Built in class :- pre-defined classes ,  
ex:- LinkedList , Scanner , Date.
- 2) User defined class :-

Code

```
public class MyClass {
    // Fields
    private int age; // not initialised
    private String name;

    // Constructor
    public MyClass(int age, String name) {
        this.age = age;
        this.name = name;
    }

    // Methods
    public void display() {
        System.out.println("age :" + age + ", name:" + name );
    }

    // main method
    public static void main(String[] args) {
        // Creating an object
        MyClass obj1 = new MyClass(25, "Manish");

        // calling a method
        obj1.display();
    }
}
```

# Code

```
public static class Movie {  
    // fields  
    String name;  
    int rating;  
    long moneyCollection;  
    int profit;  
    String actor;  
    String actress;  
  
    public Movie(String name, int rating, long moneyCollection, int profit, String actor, String actress) {  
        this.name = name;  
        this.rating = rating;  
        this.moneyCollection = moneyCollection;  
        this.profit = profit;  
        this.actor = actor;  
        this.actress = actress;  
    }  
  
    public void print() {  
        System.out.print("Hello");  
    }  
}  
  
public static void main(String[] args) {  
    Movie batman1 = new Movie("Batman1", 8, 200000, 5000, "Rachel Gupta & Nikhil Chinapa", "Disha");  
    System.out.println(batman1.rating);  
    System.out.println(batman1.moneyCollection);  
    System.out.println(batman1.profit);  
    System.out.println(batman1.actor);  
    System.out.println(batman1.actress);  
}
```