Classes, Objects → instance
→ reference

↳ blue print of creating object

↳ pulling data members and member functions together.

Person

↳ name →
↳ age →

DSA

String - class → str.sayHi()

```
public static class Person{

}
```

```
String name;
int age;
int marks;
```

→ data members

p1.sayHi()

public static class class-name {

```
void saysHi(){
    System.out.println("Hello");
}
```

↳ member fn

Declaration

```
Person p1;
```

Person → class p1 → declare

}

```
Person p1 = new Person();
```

p1.name = "Ram";
p1.age = 23;

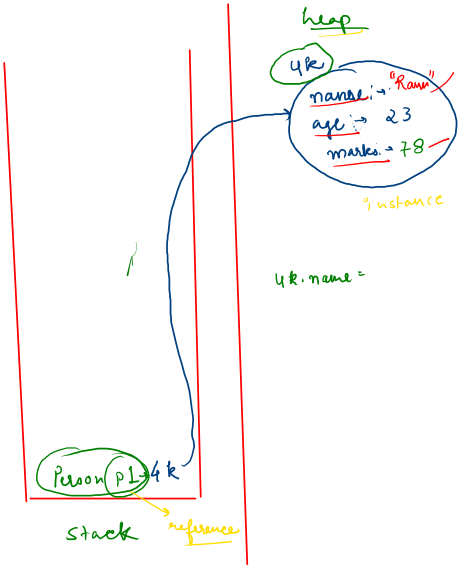
p1.marks = 78;

Person p1;

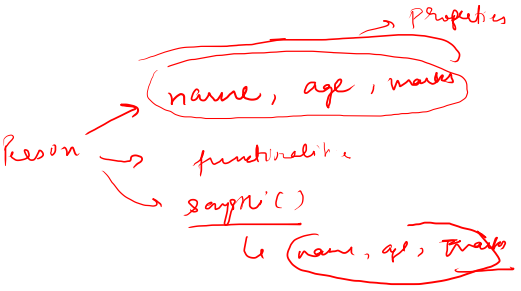
* If we don't set the object values
  then it initialize with it's default value

* To set the values → we are use dot operator

Object name dot(.) property = value;




```
public static class Person{
    String name;
    int age;
    int marks;

    void saysHi(){
        System.out.println(name +" " + age + " " + marks);
    }

}
```

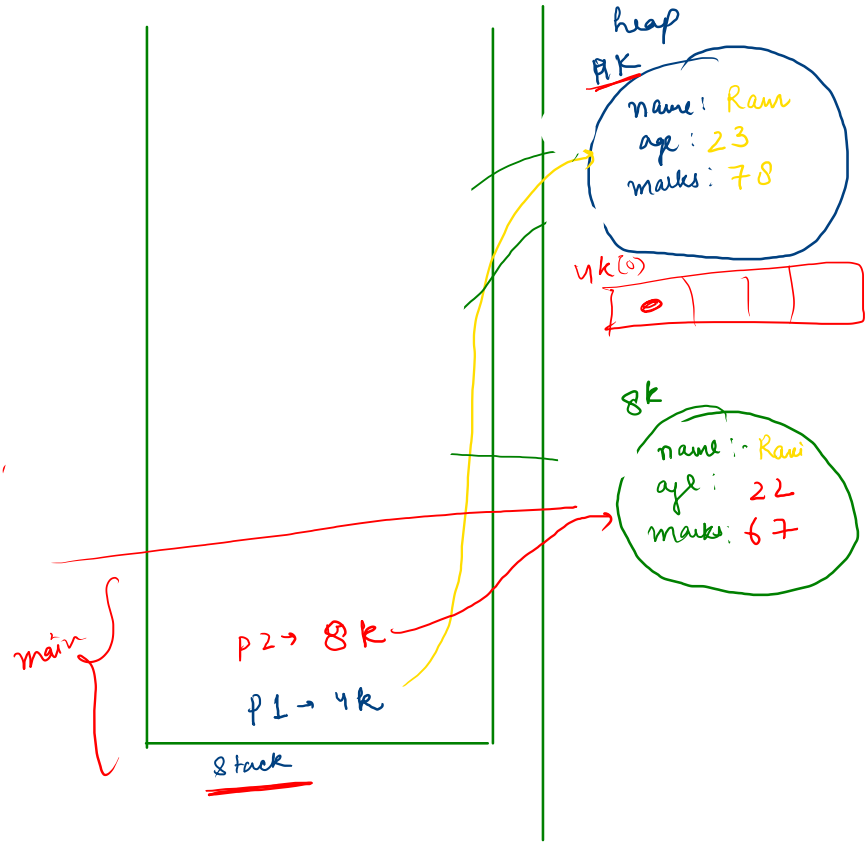


```java
public static void main(String[] args) {
    /* Enter your code here. Read input from STDIN. Print outpu

    Person p1 = new Person();
    p1.name="Ram";
    p1.age=23;
    p1.marks=78;


    Person p2 = new Person();
    p2.name="Ravi";
    p2.age=22;
    p2.marks=67;


    p1.saysHi();
    p2.saysHi();

    swap(p1,p2);


    p1.saysHi();
    p2.saysHi();

}

public static void swap(Person p1,Person p2){
    Person temp=p1;
    p1=p2;
    p2=temp;
}
```

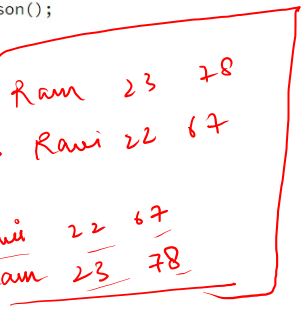
p1.saysHi(); → Ram 23 78
p2.saysHi(); → Ravi 22 67

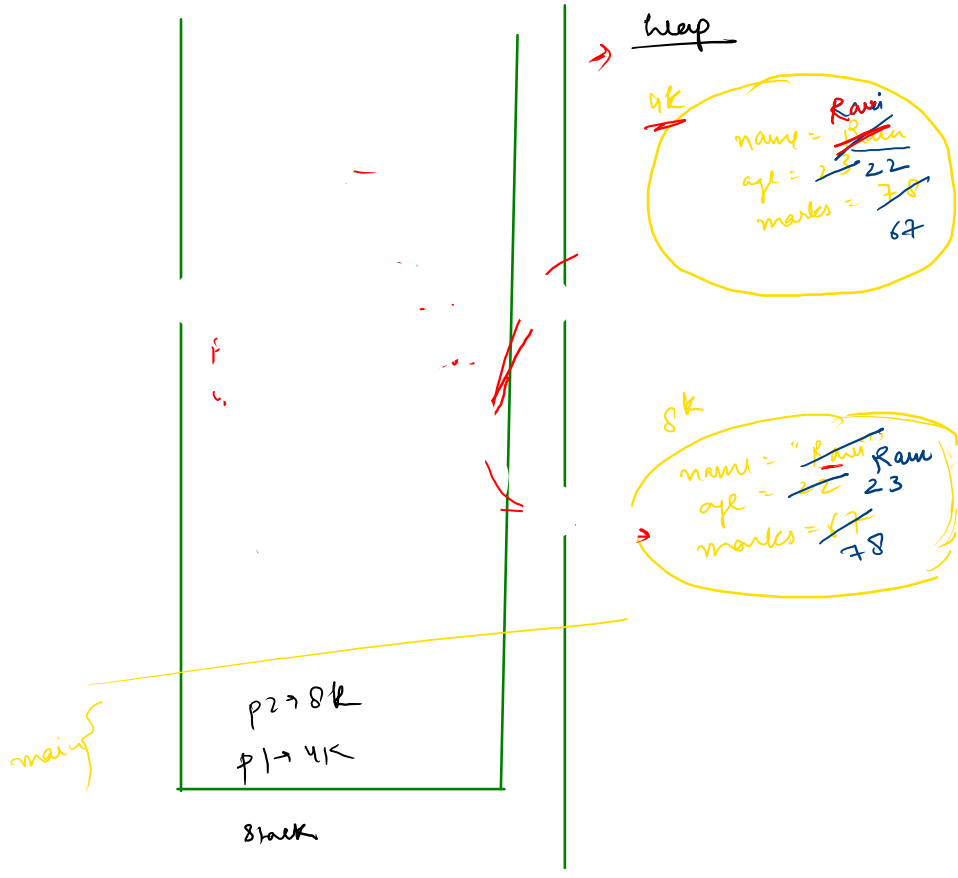
Ram 23 78
Ravi 22 67

Ravi 23 78
Ravi 22 67

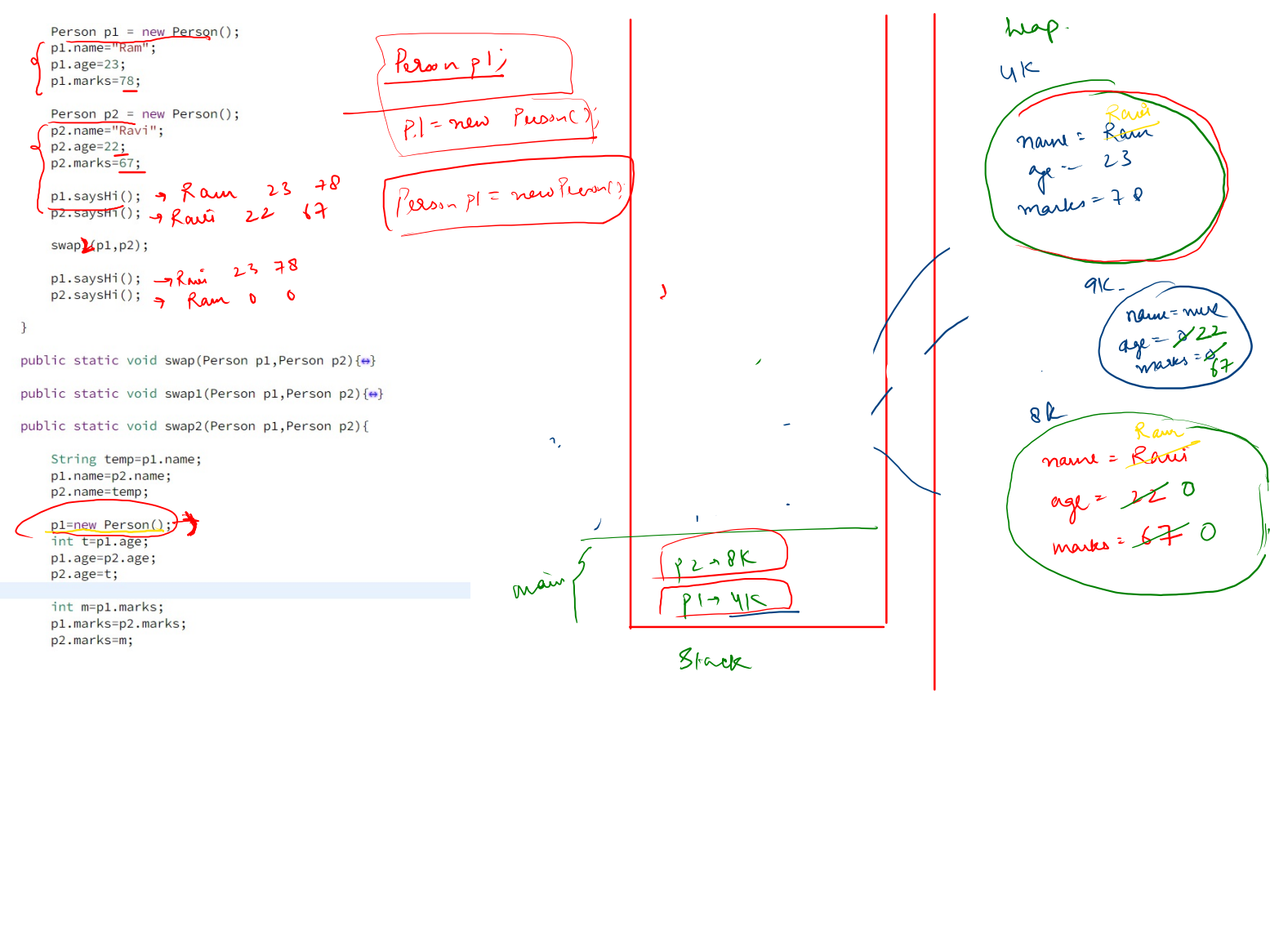


```
    Person p1 = new Person();
    p1.name="Ram";
    p1.age=23;
    p1.marks=78;


    Person p2 = new Person();
    p2.name="Ravi";
    p2.age=22;
    p2.marks=67;

    p1.saysHi();
    p2.saysHi();

    swap1(p1,p2);

    p1.saysHi();
    p2.saysHi();
}

public static void swap(Person p1,Person p2){}

public static void swap1(Person p1,Person p2){
    String temp=p1.name;
    p1.name=p2.name;
    p2.name=temp;

    int t=p1.age;
    p1.age=p2.age;
    p2.age=t;


    int m=p1.marks;
    p1.marks=p2.marks;
    p2.marks=m;
}
```





```
    Person p1 = new Person();
    p1.name="Ram";
    p1.age=23;
    p1.marks=78;

    Person p2 = new Person();
    p2.name="Ravi";
    p2.age=22;
    p2.marks=67;

    p1.saysHi();
    p2.saysHi();

    swap(p1,p2);

    p1.saysHi();
    p2.saysHi();
}

public static void swap(Person p1,Person p2){}

public static void swap1(Person p1,Person p2){}

public static void swap2(Person p1,Person p2){

    String temp=p1.name;
    p1.name=p2.name;
    p2.name=temp;

    p1=new Person();
    int t=p1.age;
    p1.age=p2.age;
    p2.age=t;

    int m=p1.marks;
    p1.marks=p2.marks;
    p2.marks=m;
```



VS Code

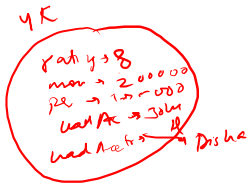
Java JDK

```java
public class Solution {

    public static class Batman{
        int rating;
        int moneyCollection;
        int profit;
        String leadActor;
        String leadActress;


        void printMovieDetails(){
            System.out.println(rating);
            System.out.println(moneyCollection);
            System.out.println(profit);
            System.out.println(leadActor);
            System.out.println(leadActress);
        }

    }
```

properties

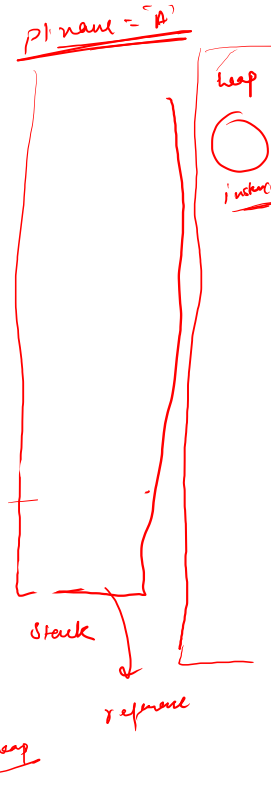
functionality

```java
public static void main(String[] args) {
    /* Enter your code here. Read input from STDIN. Print ou

    Batman batman1=new Batman();

    batman1.rating=8;
    batman1.moneyCollection=200000;
    batman1.profit=5000;
    batman1.leadActor="John";
    batman1.leadActress="Disha";


    Batman batman2=new Batman();

    batman2.rating=9;
    batman2.moneyCollection=500000;
    batman2.profit=8000;
    batman2.leadActor="Akshay";
    batman2.leadActress="Anushka";


    Batman batman3=new Batman();

    batman3.rating=10;
    batman3.moneyCollection=700000;
    batman3.profit=6000;
    batman3.leadActor="Ajay";
    batman3.leadActress="Deepika";

    batman1.printMovieDetails();
    batman2.printMovieDetails();
    batman3.printMovieDetails();

}
```



```java
public static class instaInfluencers{
    String name;
    int totalPosts;
    int totalReels;
    String blueTick;
    int followers;
    int following;
    String category;
    char gender;

    void printDetailsWithoutName(){

        System.out.println(totalPosts);
        System.out.println(totalReels);
        System.out.println(blueTick);
        System.out.println(followers);
        System.out.println(following);
        System.out.println(category);
        System.out.println(gender);
    }

    void printDetails(){
        System.out.println(name);
        System.out.println(totalPosts);
        System.out.println(totalReels);
        System.out.println(blueTick);
        System.out.println(followers);
        System.out.println(following);
        System.out.println(category);
        System.out.println(gender);
    }

}
```

```java
public static void main(String[] args) {
    /* Enter your code here. Read input from STDIN. Print output to STDOUT.


    instaInfluencers RaftaarMusic =new instaInfluencers();
    instaInfluencers NikhilChinapa =new instaInfluencers();
    instaInfluencers Baseer_bob =new instaInfluencers();

    RaftaarMusic.name="Raftaar";
    RaftaarMusic.totalPosts=340;
    RaftaarMusic.totalReels=400;
    RaftaarMusic.blueTick="Yes";
    RaftaarMusic.followers=7;
    RaftaarMusic.following=200;
    RaftaarMusic.category="Rapper";
    RaftaarMusic.gender='M';


    NikhilChinapa.name="Nikhil Chinapa";
    NikhilChinapa.totalPosts=500;
    NikhilChinapa.totalReels=220;
    NikhilChinapa.blueTick="Yes";
    NikhilChinapa.followers=5;
    NikhilChinapa.following=210;
    NikhilChinapa.category="VJ";
    NikhilChinapa.gender='M';

    Baseer_bob.name="Baseer";
    Baseer_bob.totalPosts=500;
    Baseer_bob.totalReels=120;
    Baseer_bob.blueTick="No";
    Baseer_bob.followers=2;
    Baseer_bob.following=230;
    Baseer_bob.category="VJ";
    Baseer_bob.gender='M';
        RaftaarMusic.printDetailsWithoutName();
        NikhilChinapa.printDetails();
        Baseer_bob.printDetails();
```



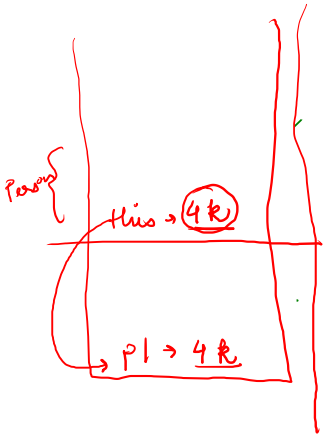
```java
import java.io.*;
import java.util.*;

public class Solution {
    public static class Movies{
       int rating;
       int moneyCollection;
       int profit;
       String leadActor;
       String leadActress;

        void printDetails(){
            System.out.println(rating);
            System.out.println(moneyCollection);
            System.out.println(profit);
            System.out.println(leadActor);
            System.out.println(leadActress);
        }

    }

    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print

        Movies Batman4=new Movies();

        Batman4.rating=9;
        Batman4.moneyCollection=700000;
        Batman4.profit=9000;
        Batman4.leadActor="Jack";
        Batman4.leadActress="Alia";

        Batman4.printDetails();
    }
}
```

```java
public static class Person{
    String name;
    int age;
    int marks;
    void saysHi(){
        System.out.println(name+" "+age+" "+marks);
    }
}

public static void main(String[] args) {
    /* Enter your code here. Read input from STDIN. Pri
    Person p1=new Person();
    p1.name="Ram";
    p1.age=23;
    p1.marks=78;

    p1.saysHi();
}
```

→ Person

p

→ default constructor



Ram 23 78

→ Parameterized Constructor

Person p1 = new Person ("Ram", 23, 78);

Person p2 = new Person ("Ram", 21, 67);

1. Space Allocation
2. Class Parsing → initialization with default values.
3. Constructor firing | Constructor calling

① ↳ special function with name similar as class name.
↳ default constructor is provided by Java.

② ↳ no return type [not even void]
↳ but by default, it returns "this"

```
public static class Movies{
    int ratings;
    int moneyCollection;
    int profit;
    String leadActor;
    String leadActress;

    public Movies(int ratings,int moneyCollection,int profit,String leadActor, String leadActress){
        this.ratings=ratings;
        this.moneyCollection=moneyCollection;
        this.profit=profit;
        this.leadActor=leadActor;
        this.leadActress=leadActress;
    }

    void printMovieDetails(){
        System.out.println(ratings);
        System.out.println(moneyCollection);
        System.out.println(profit);
        System.out.println(leadActor);
        System.out.println(leadActress);
    }


}
```

```
public static void main(String[] args) {
    /* Enter your code here. Read input from STDIN. Print output to STDOU

    Movies batman1=new Movies(8,200000,5000,"John","Disha");
    Movies batman2=new Movies(9,500000,8000,"Akshay","Anushka");
    Movies batman3=new Movies(10,700000,6000,"Ajay","Deepika");

    batman1.printMovieDetails();
    batman2.printMovieDetails();
    batman3.printMovieDetails();

}
```

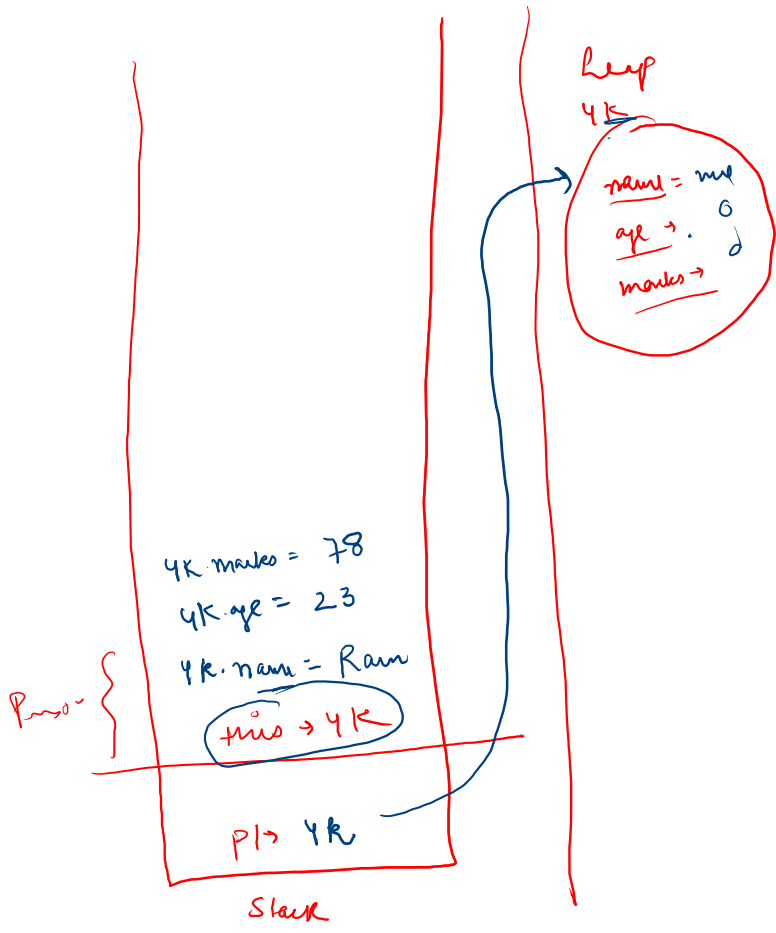
```java
public static class instaInfluencers{
    String name;
    int totalPosts;
    int totalReels;
    String blueTick;
    int followers;
    int following;
    String category;
    char gender;

    public instaInfluencers(String name,int totalPosts, int totalReels,String blueTick,int followers,int
following,String category,char gender){
        this.name=name;
        this.totalPosts=totalPosts;
        this.totalReels=totalReels;
        this.blueTick=blueTick;
        this.followers=followers;
        this.following=following;
        this.category=category;
        this.gender=gender;
    }

    void printDetailsWithoutName(){

        System.out.println(totalPosts);
        System.out.println(totalReels);
        System.out.println(blueTick);
        System.out.println(followers);
        System.out.println(following);
        System.out.println(category);
        System.out.println(gender);
    }

    void printDetails(){
        System.out.println(name);
        System.out.println(totalPosts);
        System.out.println(totalReels);
        System.out.println(blueTick);
        System.out.println(followers);
        System.out.println(following);
        System.out.println(category);
        System.out.println(gender);
    }

}

public static void main(String[] args) {
    /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be named Solutio

    instaInfluencers raftaarMusic =new instaInfluencers("Raftaar",340,400,"Yes",7,200,"Rapper",'M');
    instaInfluencers nikhilChinapa =new instaInfluencers("Nikhil Chinapa",500,220,"Yes",5,210,"VJ",'M');
    instaInfluencers baseerBob = new instaInfluencers("Baseer",500,120,"No",2,230,"VJ",'M');

    raftaarMusic.printDetailsWithoutName();
    nikhilChinapa.printDetails();
    baseerBob.printDetails();
}
```

```java
public class Solution {
    public static class Person{
        String name;
        int age;
        int marks;

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

//          default costructor
        public Person(){
            name="ram";
            age=12;
            marks=34;
        }

        void saysHi(){
            System.out.println(name+" "+age+" "+marks);
        }
    }

    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print outp

        Person p1=new Person();
        Person p2=new Person("Ravi",21,67);


        p1.saysHi();
        p2.saysHi();
    }
```

Person p1;

Person p1 = new Person();

p1("Ram", 23, 78)



Make a class named movies which has the properties as were given above.

```
Sample object batman1 of the class movies has the properties as listed below

batman1
Rating: 8
Money Collection: 200,000
Profit: 5000
Lead Actor: John
Lead Actress: Disha
```

Now make a function named propDisplay() inside this class, which displays the values of all the properties of the class movies such that each property is printed in a separate line.

Then assign the values to the object Batman1, Batman2, Batman3 as listed using the parameterized constructor.

```
public static class Movies{
    String name;
    int rating;
    String moneyCollection;
    int profit;
    String leadActor;
    String leadActress;

    public Movies(String name,int rating,String moneyCollection,int profit,String leadActor,String leadActress){
        this.name=name;
        this.rating=rating;
        this.moneyCollection=moneyCollection;
        this.profit=profit;
        this.leadActor=leadActor;
        this.leadActress=leadActress;

    }

    void propDisplay(){
        System.out.println(name);
        System.out.println(rating);
        System.out.println(moneyCollection);
        System.out.println(profit);
        System.out.println(leadActor);
        System.out.println(leadActress);
    }
}

public static void main(String[] args) {
    /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be named Solution. */
    Movies b1=new Movies("Batman 1",8,"200,000",5000,"John","Disha");
    Movies b2=new Movies("Batman 2",9,"500,000",8000,"Akshay","Anushka");
    Movies b3=new Movies("Batman 3",10,"700,000",6000,"Ajay","Deepika");

    b1.propDisplay();
    b2.propDisplay();
    b3.propDisplay();
}
```

```
public class Solution {
  public static class Movies{
      String name;
      int rating;
      String moneyCollection;
      int profit;
      String leadActor;
      String leadActress;

      public Movies(String name,int rating,String moneyCollection,int profit,String leadActor,String leadActress){
          this.name=name;
          this.rating=rating;
          this.moneyCollection=moneyCollection;
          this.profit=profit;
          this.leadActor=leadActor;
          this.leadActress=leadActress;

      }
      boolean checkHit(int rating){
          return rating>5?true:false;

          // if(rating >5) return true;
          // else return false
      }

  }

  public static void main(String[] args) {
      /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be named Solution. */
      Movies b1=new Movies("Batman 1",8,"200,000",5000,"John","Disha");
      Movies b2=new Movies("Batman 2",9,"500,000",8000,"Akshay","Anushka");
      Movies b3=new Movies("Batman 3",10,"700,000",6000,"Ajay","Deepika");

      System.out.println(b1.checkHit(b1.rating));
      System.out.println(b2.checkHit(b2.rating));
      System.out.println(b3.checkHit(b3.rating));

  }
```

Constructor Overloading

public Movies (———)

↳ String name ↳ int ratings, ↳ String moneyCollection

↳ int project ↳ String leadActor ↳ String leadActress

↳ name, moneyCollection, leadActor, leadActress → public Movies(—)

↳ project, moneyCollection → public Movies (—)

↳ Constructor

```java
public class Solution {
    public static class Movies{
        String name;
        int rating;
        int moneyCollection;
        int profit;
        String leadActor;
        String leadActress;

//          Parameterized constructor -1
        public Movies(int rating, int moneyCollection){
            this.rating =rating;
            this.moneyCollection=moneyCollection;

            System.out.println("Stored rating and money.");
        }

        //          Parameterized constructor -2
        public Movies(int rating, int moneyCollection,int profit){
            this.rating =rating;
            this.moneyCollection=moneyCollection;
            this.profit=profit;

            System.out.println("Stored rating, money, profit are given.");
        }


        //          Parameterized constructor -3
        public Movies(int rating, int moneyCollection,int profit,String leadActor){
            this.rating =rating;
            this.moneyCollection=moneyCollection;
            this.profit=profit;
            this.leadActor=leadActor;

            System.out.println("Stored rating, money, profit, lead actors are given.");
        }

    }

    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your
        Movies Superman1=new Movies(8,900000);
        Movies Superman2=new Movies(8,900000,8000);
        Movies Superman3=new Movies(8,900000,8000,"Ranveer Singh");
    }
```

```
public static class Movies{
    String name;
    int rating;
    int moneyCollection;
    int profit;
    String leadActor;
    String leadActress;


    public Movies(int rating,int moneyCollection,int profit,String leadActor,String leadActress){
        this.rating=rating;
        this.moneyCollection=moneyCollection;
        this.profit=profit;
        this.leadActor=leadActor;
        this.leadActress=leadActress;

        System.out.println("Lead actor has the main role.");
    }

    public Movies(String leadActress,String leadActor,int rating,int moneyCollection,int profit){
        this.leadActress=leadActress;
        this.leadActor=leadActor;
        this.rating=rating;
        this.moneyCollection=moneyCollection;
        this.profit=profit;


        System.out.println("Lead actress has the main role.");

    }

}
public static void main(String[] args) {
    /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class
    Movies Superman1=new Movies(8,900000,8000,"Ranveer Singh","Katrina Kaif");
    Movies Superman2=new Movies("Vani Gupta", "Raj Gupta",9,1900000,28000);
}
```

Static Key word

Static

"XYZ"

Advantages

↳ Memory efficient → ?

↳ for static, we don't need an object → ?

Students. school-name;

or

s1. school name;

s2. school name;

Non Static

Students s1 = new Students('A', 1, "XYZ");

Students s2 = new Students('B', 2, "XYZ");

Disadvantages → wastage of memory → ?

s1. school... s1 4K

Name → 'A'
Roll no = 1
school-name = 'XYZ'

s1. name = null

s2

name = null
roll = 0
school = XYZ

5K

Name - 'B'
Roll no - 2
schoolname = 'XYZ'

5 - A

class → Students

{

String student-name;

int roll-no;

static String school-name; "XYZ"

}

Students. schoolName = "XYZ";

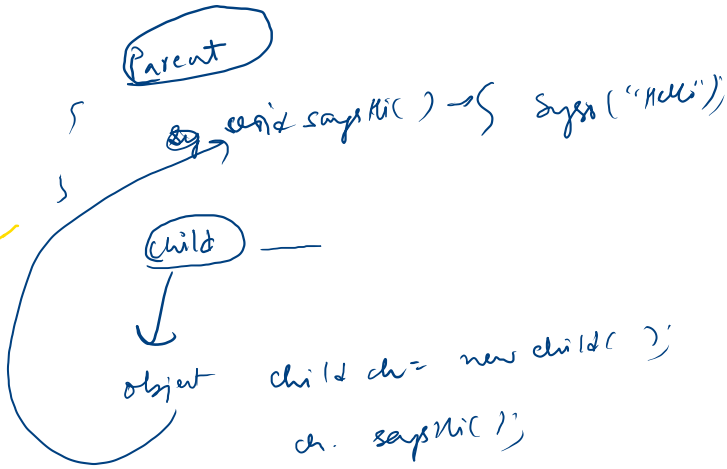
```
    public static class Students{
        String studentName;
        int rollNo;
        static String schoolName;

        static{  // done when we load a class
            schoolName="XYZ";
            System.out.println("We are in static block");
        }

        public Students(String studentName,int rollNo){ // Done when we create an object
            this.studentName=studentName;
            this.rollNo=rollNo;

            System.out.println("We are in constructor");
        }

        void printDetails(){
            System.out.println(studentName+" "+rollNo+" "+schoolName);
        }
    }

public static void main(String[] args) {
    /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should

    Students s1=new Students("A",1);
    Students s2=new Students("B",2);
    Students.schoolName="XYZ";
    System.out.println(s1.schoolName);
    System.out.println(s2.schoolName);

    // s1.printDetails();
    // s2.printDetails();


    Students.schoolName="ABC"; // For static we don't need object;

    // System.out.println(Students.schoolName);

    // s1.printDetails();
    // s2.printDetails();
}
}
```

```java
public class Solution {
      public static class Rectangle{
          int length;
          int breadth;

          public Rectangle(int length,int breadth){
              this.length=length;
              this.breadth=breadth;
          }

           int area(int length,int breadth){
              return length*breadth;
          }

      }

    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class sh
        Rectangle rect=new Rectangle(10,5);
        System.out.println(rect.area(10,5));

    }
}
```

```
public static class Movies{
    String name;
    int ratings;
    int moneyCollection;
    int profit;
    String leadActor;
    String leadActress;
    static String videoType;


    // static {
    //     videoType="movies";
    // }

    public Movies(int ratings,int moneyCollection,int profit,String leadActor,String leadActress){
        this.ratings=ratings;
        this.moneyCollection=moneyCollection;
        this.profit=profit;
        this.leadActor=leadActor;
        this.leadActress=leadActress;

    }
}


public static void main(String[] args) {
    /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be named Solution. */
    Movies Superman1=new Movies(8,900000,8000,"Ranveer Singh","Katrina Kaif");
    Movies.videoType="movies";
    System.out.println(Superman1.videoType);
    System.out.println(Movies.videoType);

}
```

Inheritance

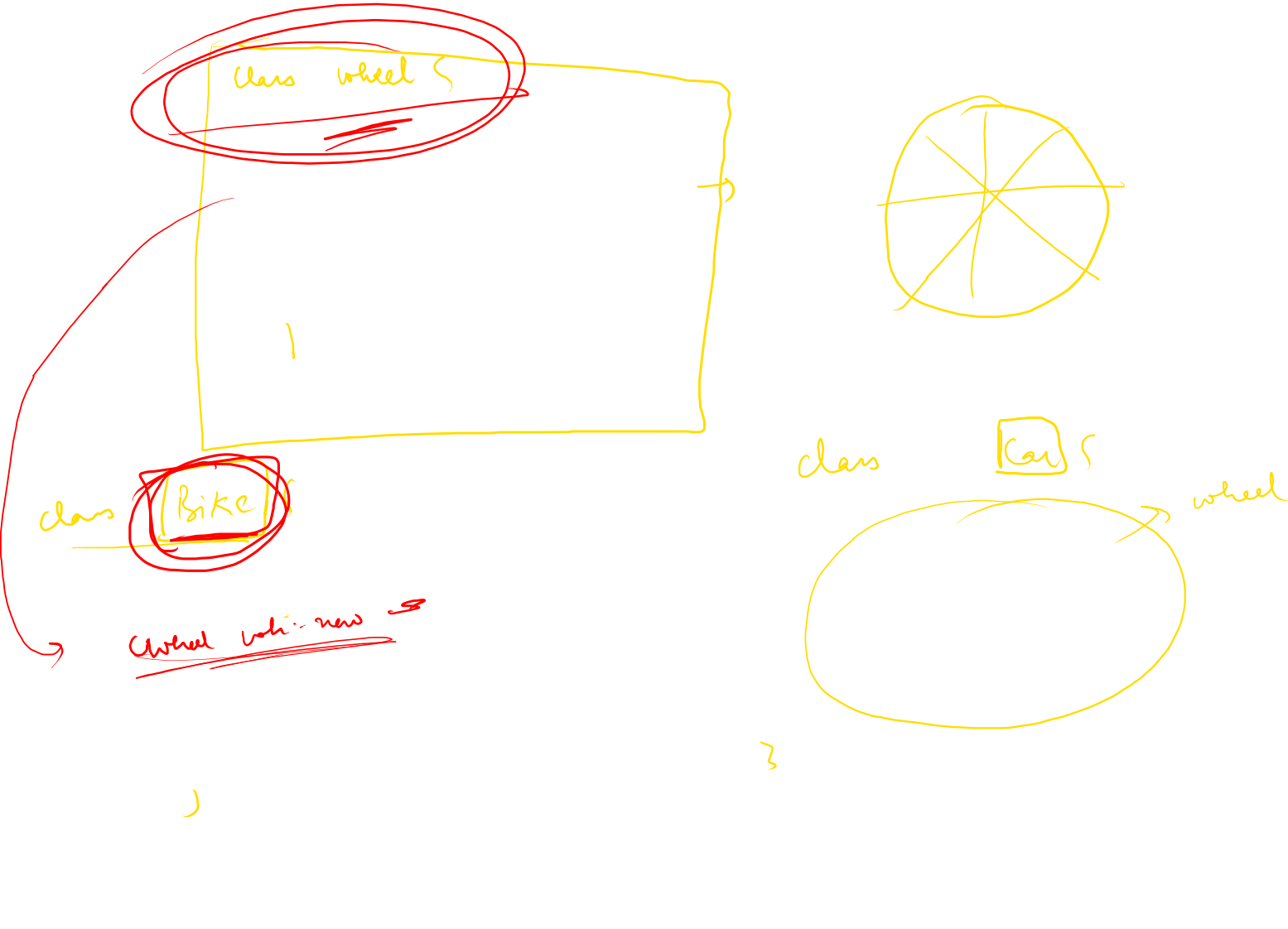
↳ It is a mechanism in which we acquire the properties & methods of one class into another.



child
{
sayHi()
{ Sysout("Hello");
}
}

↳ Code Reusability

↳ ? Method Overriding


class wheel {
}
class Bike {
wheel obj = new
class Car {
wheel

Superclass / Parent class → Are those classes whose features are inherited;

Ex → wheel (Parent)

Subclass / Child class → Are those classes that inherits the other class.

Ex → Bike, car (Child)

Keyword → extends

```
class wheel {
}
```

```
class Bike extends wheel{
}
```

```
class Car extends wheel{
}
```

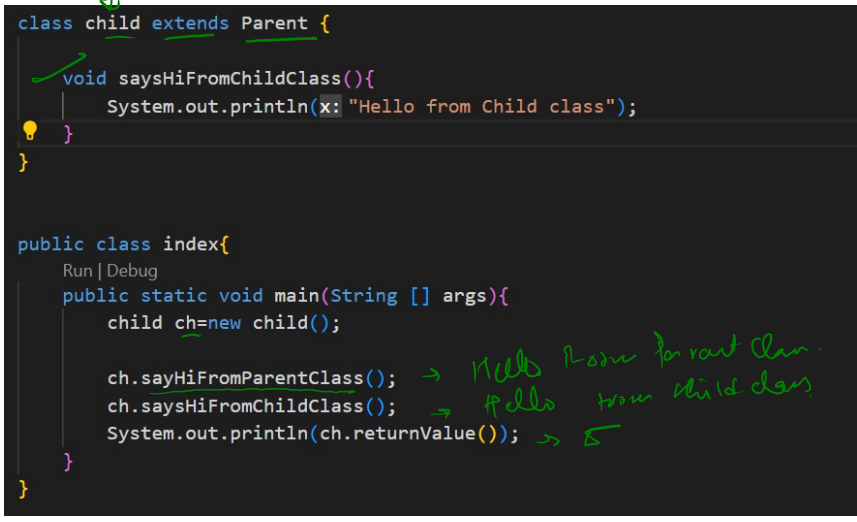
Syntax :

```
class subclass extends superclass {
}
```

subclass → child

superclass → parent


class Parent{
void sayHiFromParentClass(){
System.out.println(x: "Hello from Parent class");
}
int returnValue(){
return 5;
}
}


class child extends Parent {
void saysHiFromChildClass(){
System.out.println(x: "Hello from Child class");
}
}
public class index{
Run | Debug
public static void main(String [] args){
child ch=new child();
ch.sayHiFromParentClass();
ch.saysHiFromChildClass();
System.out.println(ch.returnValue());
}
}

Constructor in Inheritance

```java
class Parent{
    void sayHiFromParentClass(){
        System.out.println(x: "Hello from Parent class");
    }
    int returnValue(){
        return 5;
    }
}
```

```java
class child extends Parent {
    void saysHiFromChildClass(){
        System.out.println(x: "Hello from Child class");
    }
}


public class index{
    Run | Debug
    public static void main(String [] args){
        child ch=new child();

        ch.sayHiFromParentClass();
        ch.saysHiFromChildClass();
        System.out.println(ch.returnValue());
    }
}
```

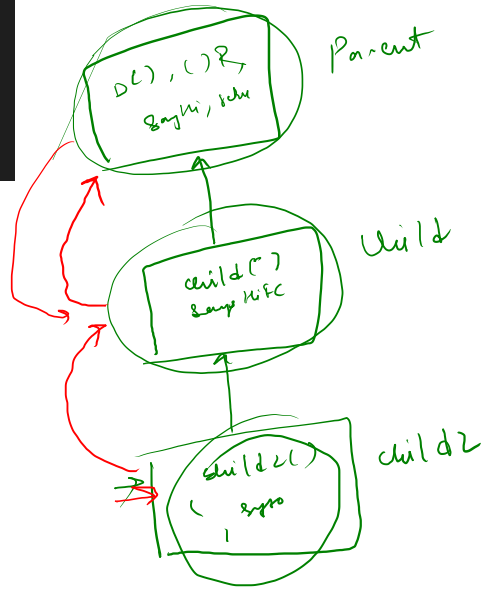
Note) - When a child class is extended from parent class, the constructor of (default) Parent class is executed first followed by constructor of child class (default).

```java
class Parent{
    public Parent(){
        System.out.println(x: "I am a constructor in Parent Class");
    }

    void sayHiFromParentClass(){
        System.out.println(x: "Hello from Parent class");
    }

    int returnValue(){
        return 5;
    }
}
```

```java
class child extends Parent {
    public child(){
        System.out.println(x: "I am a constructor in Child Class");
    }

    public child (int a){
        System.out.println(a);
    }

    void saysHiFromChildClass(){
        System.out.println(x: "Hello from Child class");
    }
}
```

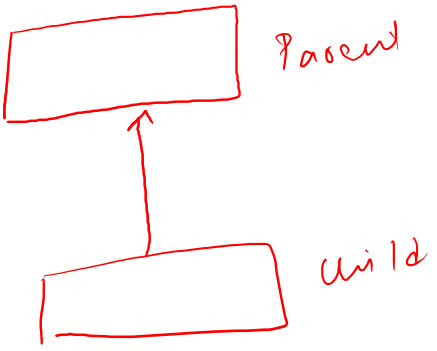
```java
class child2 extends child {
    public child2(){
        System.out.println(x: "I am a constructor in Child2 Class");
    }


}


public class index{
    Run | Debug
    public static void main(String [] args){
        child2 ch=new child2();

        // ch.sayHiFromParentClass();
        // ch.saysHiFromChildClass();
        // System.out.println(ch.returnValue());
    }
}
```

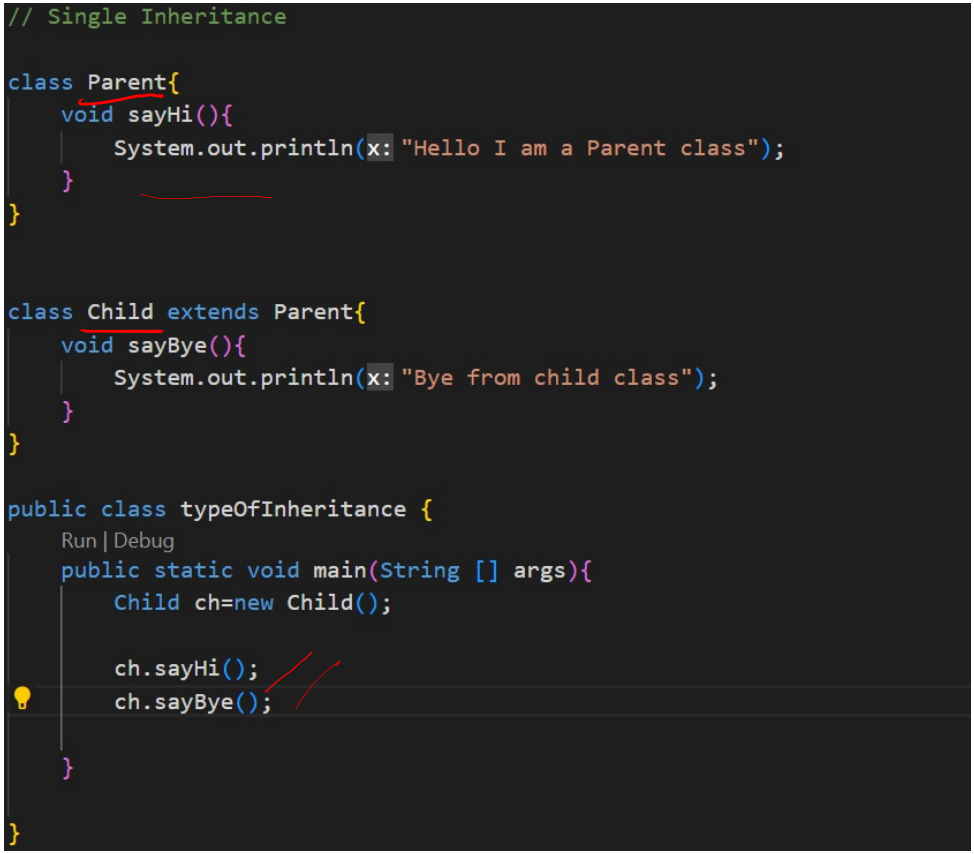
```
I am a constructor in Parent Class
I am a constructor in Child Class
I am a constructor in Child2 Class
```



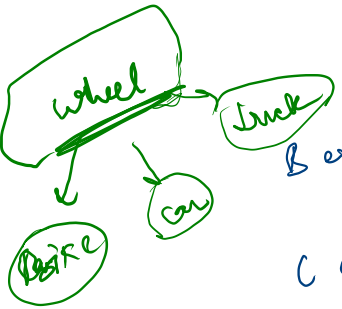
Types of Inheritance

1. Single Inheritance



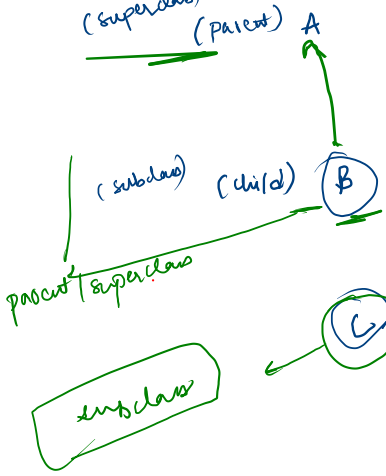

// Single Inheritance
class Parent{
void sayHi(){
System.out.println(x: "Hello I am a Parent class");
}
}
class Child extends Parent{
void sayBye(){
System.out.println(x: "Bye from child class");
}
}
public class typeOfInheritance {
Run | Debug
public static void main(String [] args){
Child ch=new Child();
ch.sayHi();
ch.sayBye();
}
}

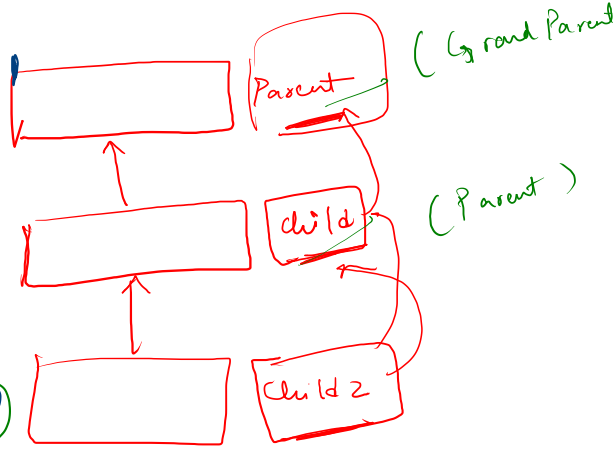
1 Multilevel Inheritance

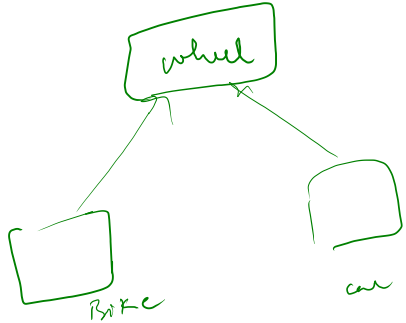


B extends A

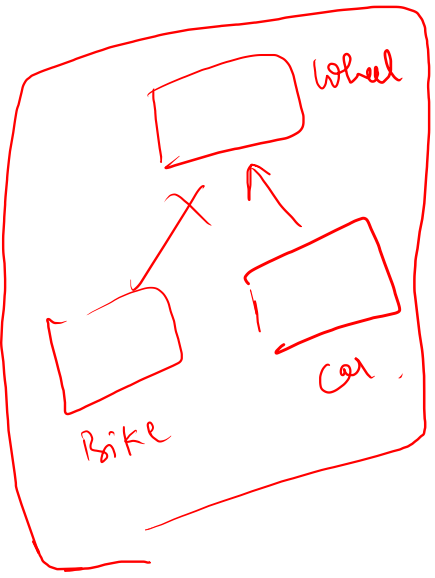
C extends B

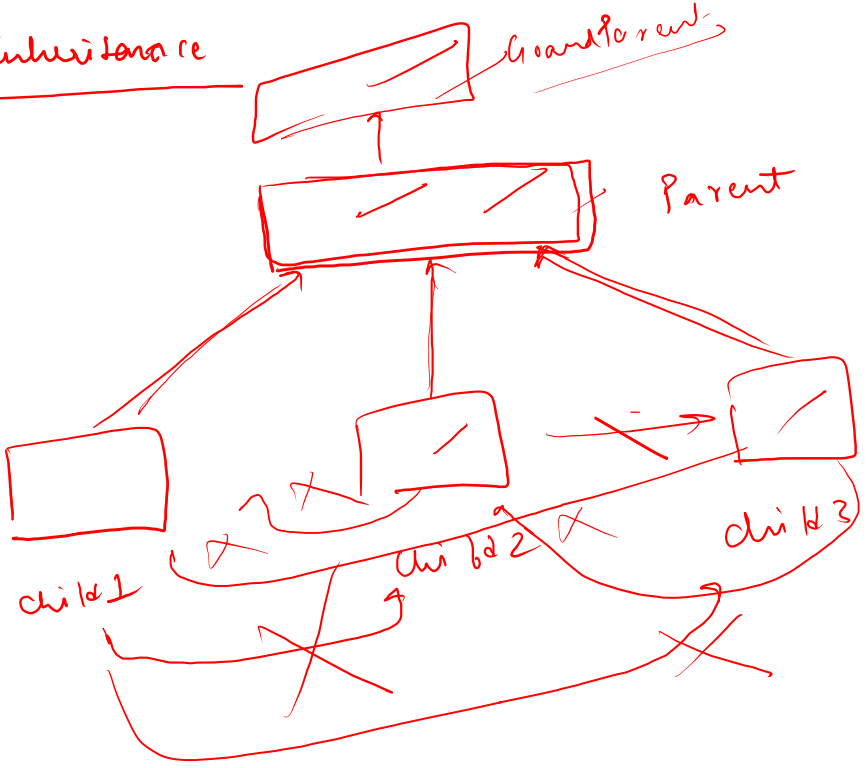






3 Hierarchical Inheritance





```java
class Parent{
    void sayHi(){
        System.out.println(x: "Hello I am a Parent class");
    }
}


class Child1 extends Parent{
    void sayBye(){
        System.out.println(x: "Bye from child1 class");
    }
}


class Child2 extends Parent{
    void sayBye2(){
        System.out.println(x: "Bye from child2 class");
    }
}

class Child3 extends Parent{
    void sayBye3(){
        System.out.println(x: "Bye from child3 class");
    }
}
```

```java
public class typeOfInheritance {
    Run | Debug
    public static void main(String [] args){
        Child1 ch=new Child1();

        ch.sayHi();
        ch.sayBye();

        Child2  ch2=new Child2();

        ch2.sayHi();
        ch2.sayBye2();


        Child3  ch3=new Child3();

        ch3.sayHi();
        ch3.sayBye3();

    }
}
```

```java
class Influencers{
    String name;
    int totalPosts;
    int totalReels;
    boolean blueTick;
    int followers;
    int following;
    String category;
    char gender;

    public void printDetails(){
        System.out.println(name);
        System.out.println(totalPosts);
        System.out.println(totalReels);
        System.out.println(blueTick);
        System.out.println(followers);
        System.out.println(following);
        System.out.println(category);
        System.out.println(gender);
    }

}
```

```java
public class Solution {

    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print
        Scanner scn=new Scanner(System.in);
        String object=scn.nextLine();
        String name=scn.nextLine();
        int tP=scn.nextInt();
        int tR=scn.nextInt();
        boolean bT=scn.nextBoolean();
        int fL=scn.nextInt();
        int fO=scn.nextInt();
        String ca=scn.next();
        char g=scn.next().charAt(0);

        Influencers obj=new Influencers();
        obj.name=name;
        obj.totalPosts=tP;
        obj.totalReels=tR;
        obj.blueTick=bT;
        obj.followers=fL;
        obj.following=fO;
        obj.category=ca;
        obj.gender=g;

        obj.printDetails();

    }
}
```

```java
class Influencers{
    String name;
    int totalPosts;
    int totalReels;
    String blueTick;
    int followers;
    int following;
    String category;
    char gender;

    public Influencers(String name,int tP,int tR,String bT,int fL,int fO,String ca,char g){
        this.name=name;
        this.totalPosts=tP;
        this.totalReels=tR;
        this.blueTick=bT;
        this.followers=fL;
        this.following=fO;
        this.category=ca;
        this.gender=g;
    }

    public void printDetails(){
        System.out.println(name);
        System.out.println(totalPosts);
        System.out.println(totalReels);
        System.out.println(blueTick);
        System.out.println(followers);
        System.out.println(following);
        System.out.println(category);
        System.out.println(gender);
    }

}
```

```java
public class Solution {

    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your cl
        Scanner scn=new Scanner(System.in);
        String object=scn.nextLine();
        String name=scn.nextLine();
        int tP=scn.nextInt();
        int tR=scn.nextInt();
        String bT=scn.next();
        int fL=scn.nextInt();
        int fO=scn.nextInt();
        String ca=scn.next();
        char g=scn.next().charAt(0);

        Influencers obj=new Influencers(name,tP,tR,bT,fL,fO,ca,g);

        obj.printDetails();


    }
}
```

```java
class Influencers{
    String name;
    int totalPosts;
    int totalReels;
    String blueTick;
    int followers;
    int following;
    String category;
    char gender;

    public Influencers(String name,int tP,int tR,String bT,int fL,int fO,String ca,char g){
        this.name=name;
        this.totalPosts=tP;
        this.totalReels=tR;
        this.blueTick=bT;
        this.followers=fL;
        this.following=fO;
        this.category=ca;
        this.gender=g;
    }

    public String checkVerified(){
        if(blueTick.equals("Yes")){
            return "Already Verified";
        }else {
            return "Needs Verification";
        }
    }

}
```

```java
public class Solution {

    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to STDO
        Scanner scn=new Scanner(System.in);
        String object=scn.nextLine();
        String name=scn.nextLine();
        int tP=scn.nextInt();
        int tR=scn.nextInt();
        String bT=scn.next();
        int fL=scn.nextInt();
        int fO=scn.nextInt();
        String ca=scn.next();
        char g=scn.next().charAt(0);

        Influencers obj=new Influencers(name,tP,tR,bT,fL,fO,ca,g);

        System.out.println(obj.checkVerified());


    }
}
```

Method Overriding :-

If a child class implements same method as present in parent, then method of child runs.

This is called method overriding.

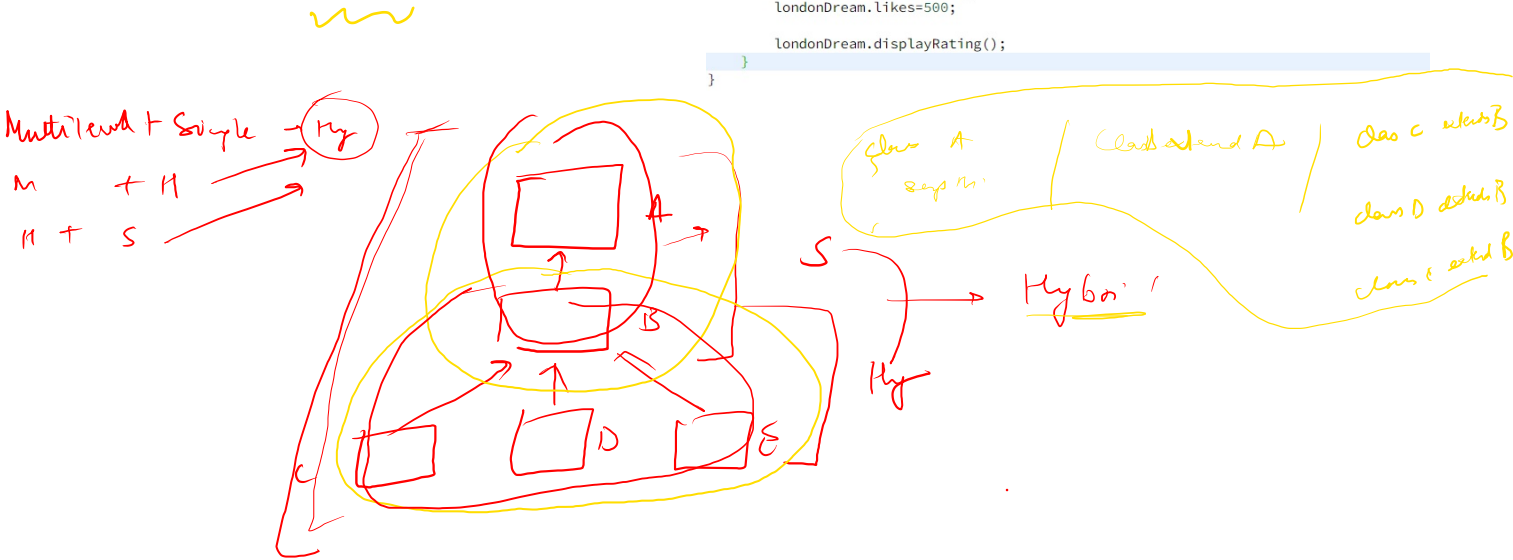
```java
class Parent1{
    public void saysHi(){
        System.out.println(x: "Hi from Parent");
    }

    int returnValue(){
        return 5;
    }
}
```

```java
class children extends Parent1{

    public void saysHi(){
        System.out.println(x: "Hi from Child");
    }
}

public class overridding{
    Run | Debug
    public static void main(String [] args){
        children ch=new children();
        Parent1 p=new Parent1();
        p.saysHi();
        ch.saysHi();
        System.out.println(ch.returnValue());

    }
}
```

```java
import java.io.*;
import java.util.*;

class Movies{
    int rating;
    int moneyCollection;
    int profit;
    String leadActor;
    String leadActress;

    public void displayRating(){
        System.out.println(rating);
        System.out.println("This is the function of movies class.");
    }
}

class CommercialMovies extends Movies{
    int views;
    int likes;


}

public class Solution {

    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class

        CommercialMovies londonDream =new CommercialMovies();
        londonDream.rating =10;
        londonDream.moneyCollection=9000000;
        londonDream.profit=70000000;
        londonDream.leadActor="Salman Khan";
        londonDream.leadActress="Asin";
        londonDream.views=10000;
        londonDream.likes=500;

        londonDream.displayRating();
    }
}
```

```java
import java.io.*;
import java.util.*;

class Movies{
    int rating;
    int moneyCollection;
    int profit;
    String leadActor;
    String leadActress;

    public void displayRating(){
        System.out.println(rating);
        System.out.println("This is the function of movies class.");
    }
}
```

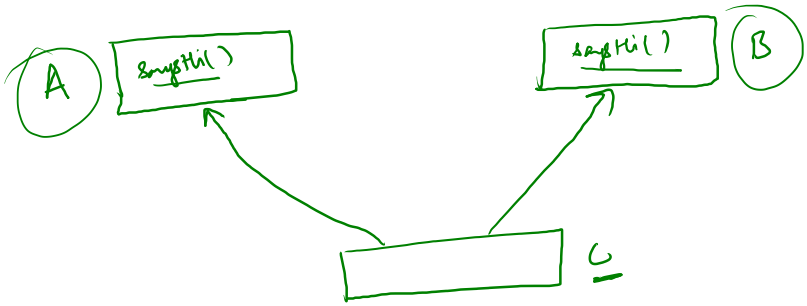
```java
class CommercialMovies extends Movies{
    int views;
    int likes;


    public void displayRating(){
        System.out.println(rating);
        System.out.println("This is the function of the commercial movies class.");
    }


}

public class Solution {

    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your cl

        CommercialMovies londonDream =new CommercialMovies();
        londonDream.rating =10;
        londonDream.moneyCollection=9000000;
        londonDream.profit=70000000;
        londonDream.leadActor="Salman Khan";
        londonDream.leadActress="Asin";
        londonDream.views=10000;
        londonDream.likes=500;

        londonDream.displayRating();
    }
}
```

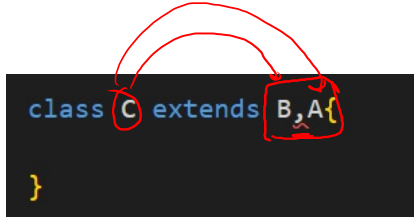


```java
import java.io.*;
import java.util.*;

class Movies{
    String name;
    int rating;
    int moneyCollection;
    int profit;
    String leadActor;
    String leadActress;

    public void displayRating(){
        // System.out.println(rating);
        System.out.println("This is the function of movies class.");
    }
}

class CommercialMovies extends Movies{
    int views;
    int likes;

    public void displayRating(){
        // System.out.println(rating);
        System.out.println("This is the function of the commercial movies class.");
    }


}

class CreativeMovies extends Movies{
    int views;
    int likes;

    public void displayRating(){
        // System.out.println(rating);
        System.out.println("Inside the creativeMovies class");
    }
}

public class Solution {

    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class
        CommercialMovies londonDreams=new CommercialMovies();

        londonDreams.name="London Dreams";
        londonDreams.rating=10;
        londonDreams.moneyCollection=9000000;
        londonDreams.profit=70000000;
        londonDreams.leadActor="Salman Khan";
        londonDreams.leadActress="Asin";
        londonDreams.views=10000;
        londonDreams.likes=500;

         CreativeMovies andhadhun=new CreativeMovies();

         andhadhun.name="Andhadhun";
         andhadhun.rating=10;
         andhadhun.moneyCollection=7000000;
         andhadhun.profit=800000;
         andhadhun.leadActor="Ayushman Khurana";
         andhadhun.leadActress="Disha";
         andhadhun.views=20000;
         andhadhun.likes=400;

         londonDreams.displayRating();
         andhadhun.displayRating();
```

Multiple Inheritance

?





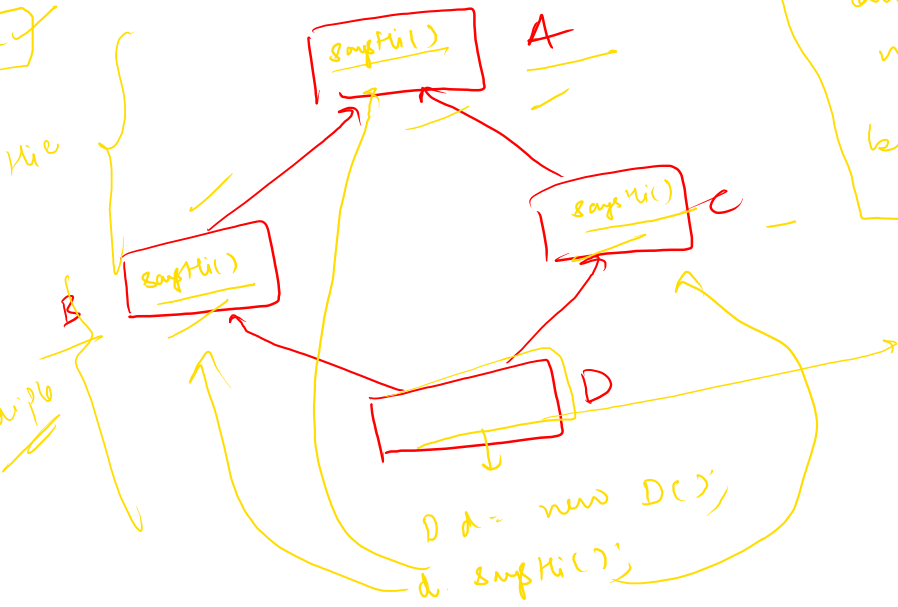

```java
class A{
    public void saysHi(){
        System.out.println(x: "Hi from class A");
    }
}

class B{
    public void saysHi(){
        System.out.println(x: "Hi fomr class B");
    }
}
```

```java
public class multipleInh {
    public static void main(String[] args){
        C ch=new C();
        ch.saysHi();
    }
}
```

Diamond Problem

((class)

is Interface



Note:

Through class in java, we can't achieve multiple inheritance.

but through interface we can achieve it

Access Modifiers

↳ accessibility or scope of a property, methods, class.

↳ (1) Private

↳ (2) Default

↳ (3) Protected

↳ (4) Public

private :- It is accessible only inside the class.

default → It is accessible in the same package.

protected → ——— × ——— but it is accessible in other package also through inheritance.

public → same package, different package

Access → / Modifier ↓	Within class	Same package	Different package
private	✓	✗	✗
default	✓	✓	✗
protected	✓	✓	✓ (Subclass) (condition: through inheritance)
public	✓	✓	✓

```java
class Movies{
    public String name;
    public int rating;
    public int moneyCollection;
    public int profit;
    public String leadActor;
    public String leadActress;
    private int views;
    private int likes;

    public Movies(String name,int rating,int moneyCollection,int profit, String leadActor,String leadActress,int views,int
likes){
        this.name=name;
        this.rating=rating;
        this.moneyCollection=moneyCollection;
        this.profit=profit;
        this.leadActor=leadActor;
        this.leadActress=leadActress;
        this.views=views;
        this.likes=likes;
    }
     public void displayViews(){
         System.out.println(views);
     }

     public void displayLikes(){
         System.out.println(likes);
     }
}
public class Solution {

     public static void main(String[] args) {
         /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be na

         Movies superman1=new Movies("Superman 1",8,90000,1000,"Ranveer","Katrina",10000,500);
         System.out.println(superman1.rating);
         System.out.println(superman1.moneyCollection);
         System.out.println(superman1.profit);
         System.out.println(superman1.leadActor);
         System.out.println(superman1.leadActress);


         superman1.displayViews();
         superman1.displayLikes();
     }
}
```

public void setter(int views){

this.views = views;

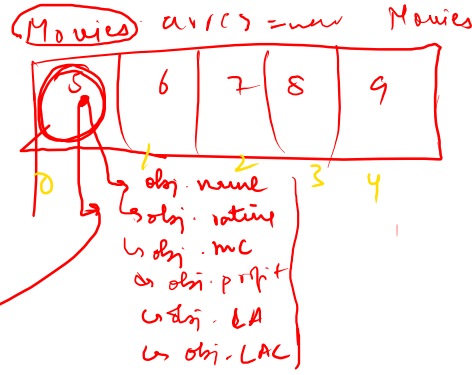
```
class Movies{
    public String name;
    public int rating;
    public int moneyCollection;
    public int profit;
    public String leadActor;
    public String leadActress;
    private int views;
    private int likes;

    public Movies(String name,int rating,int moneyCollection,int profit, String leadActor,String leadActress,int views,int
likes){
        this.name=name;
        this.rating=rating;
        this.moneyCollection=moneyCollection;
        this.profit=profit;
        this.leadActor=leadActor;
        this.leadActress=leadActress;
        this.views=views;
        this.likes=likes;
    }

    public int  displayViews(){
        return views;
    }

    public int displayLikes(){
         return likes;
    }

    private int calculateGrade(int views,int likes){
        int finalGrade=10*likes+views;
        return finalGrade;
    }

    public int printGrade(int likes,int views){
        int ans=calculateGrade(views,likes);
        return ans;
    }
}
```

```
public class Solution {

    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be na

        Movies superman1=new Movies("Superman 1",8,90000,1000,"Ranveer","Katrina",10000,500);

        int views=superman1.displayViews();
        int likes=superman1.displayLikes();

        int ans=superman1.printGrade(likes,views);
        System.out.println(ans);


    }
}
```

T - B

```
class students {

    student.name

    ~~student.school.name~~

}
```

static student.school.name = "ABC";




```java
public class Solution {

    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be named Solution. */
        Scanner scn=new Scanner(System.in);

        int n=scn.nextInt();
        Movies arr[]=new Movies[n];
        // int arr[]=new int[n];

        for(int i=0;i<n;i++){
            String name=scn.next();
            int rating =scn.nextInt();
            int moneyCollection=scn.nextInt();
            int profit=scn.nextInt();
            String leadActor=scn.next();
            String leadActress=scn.next();

            Movies obj=new Movies(name,rating,moneyCollection,profit,leadActor,leadActress);
            arr[i]=obj;
        }

        for(int i=0;i<n;i++){

            Movies obj=arr[i];
            // System.out.println(arr[i].name);
            // System.out.println(arr[i].rating);
            // System.out.println(arr[i].moneyCollection);
            // System.out.println(arr[i].profit);
            // System.out.println(arr[i].leadActor);
            // System.out.println(arr[i].leadActress);

             System.out.println(obj.name);
            System.out.println(obj.rating);
            System.out.println(obj.moneyCollection);
            System.out.println(obj.profit);
            System.out.println(obj.leadActor);
            System.out.println(obj.leadActress);
        }

    }
```

```java
import java.io.*;
import java.util.*;

class Movies{
    public String name;
    public int rating;
    public int moneyCollection;
    public int profit;
    public String leadActor;
    public String leadActress;


    public Movies(String name,int rating,int moneyCollection,int profit, String leadActor,String leadActress){
        this.name=name;
        this.rating=rating;
        this.moneyCollection=moneyCollection;
        this.profit=profit;
        this.leadActor=leadActor;
        this.leadActress=leadActress;

    }

}
```

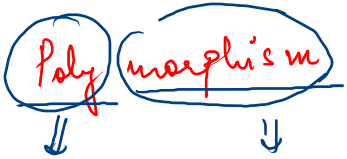
Encapsulation →

Binding of properties and methods as a single unit

→ It can be achieved by making the properties private

```java
class EmployeeDetails{
    private String  empName;
    private int empId;

    // Getter and setter
    public int getEmpId(){
        return empId;
    }

    public void setEmpDetails(int Id,String name){
        empId = Id;
        this.empName = name;
    }
    public void setEmpName(String name){
        this.empName = name;
    }

    public String getEmpName(){
        return empName;
    }
}
```

```java
public class Demo {

    Run | Debug
    public static void main(String []args){
        EmployeeDetails e=new EmployeeDetails();
        // e.empId=5;
        e.setEmpDetails(Id: 5,name: "Akash");
        System.out.println(e.getEmpId());

        // e.setEmpName("Akash");
        System.out.println(e.getEmpName());

    }

}
```



many + forms = many forms.

(U) It is defined as the ability of a message to be displayed in many forms.

It is of 2 types

1. Compile-time polymorphism | Static polymorphism

- It is achieved by Method overloading

class Students {

void sayHi()

void sayHi(name)

void sayHi(value)

void sayHi(name, value);

}

String sayHi()
{ return "Hi"; }

int sum(a, b);

int sum(a, b, c);

int sum(a, b, c, d);

float sum(a, b);

float sum(a, b, c);

```java
class Cal{

    int sum(int a,int b){
        return a+b;

    }

    int sum(int a,int b,int c){
        return a+b+c;
    }

    int sum(int a, int b,int c,int d){
        return a+b+c+d;
    }

    long sum(long a,long b){
        return a+b;
    }

    double sum(double a,double b){
        return a+b;
    }

}
```

```java
public class Demo {
    Run | Debug
    public static void main(String []args){
        Cal c=new Cal();


        System.out.println(c.sum(a: 10000000000L,b: 3123123100L));
        System.out.println(c.sum(a: 2,b: 3));
        System.out.println(c.sum(a: 1,b: 2,c: 4));
        System.out.println(c.sum(a: 12.23,b: 34.45));

    }

}
```

② Run-time polymorphism

Dynamic polymorphism

Method Overloading

Method Overriding

↳ Inheritance

↳ By Method Overriding

Parent

```
{
    saystri()
    {
    }
}
```

child extends Parent {

super.saystri()

saystri()

{

}

```java
class Parent{

    void saysHi(){
        System.out.println(x: "Hi from Parent");
    }
}


class Child extends Parent{

    void saysHi(){
        System.out.println(x: "Hi from Child");
    }
}
```

```java
public class Demo {
    Run | Debug
    public static void main(String []args){
        // Cal c=new Cal();


        // System.out.println(c.sum(10000000000L,3123123100L));
        // System.out.println(c.sum(2,3));
        // System.out.println(c.sum(1,2,4));
        // System.out.println(c.sum(12.23,34.45));


        Child ch=new Child();

        ch.saysHi();
    }

}
```

Abstraction

↳ hiding implementation

↳ achieved by 2 ways.

1. Abstract class → (0 - 100)%
2. Interface → (100%)

Encapsulation

↳ hiding data/information

```java
abstract class Vehicle{
    abstract void start();

    void saysHI(){
        System.out.println(x: "HI");
    }
}

class Car extends Vehicle{

    void start(){
        System.out.println(x: "Start with key");
    }
}

class Bike extends Vehicle{

    void start(){
        System.out.println(x: "Start with self and kick");
    }
}
```

```java
public class Demo {

    Run | Debug
    public static void main(String []args){
        Car c=new Car();
        c.start();
        c.saysHI();

        Bike b = new Bike();
        b.start();
        b.saysHI();

        Vehicle v;


    }
}
```

② Interface :-

↳ they are similar to abstract class but having all methods abstracts.

↳ They are the blueprint of class which specify what a class can do but not how.

Advantages

↳ It supports multiple inheritance.

↳ Abstraction

interface interface-name {

void stack ();

}

↳ implements

```java
interface A{
    void start();

    void saysHi();

}

interface B{
    void start();

    void saysHi();
}
```

```java
class C implements A,B{
    public void start(){
        System.out.println(x: "HI, I'm from Interface");
    }
    public void saysHi(){
        System.out.println(x: "Hi from interface ");
    }

}
```

```java
public class Demo {

    Run | Debug
    public static void main(String []args){
        C c=new C();
        c.start();
        c.saysHi();


    }

}
```

(1) Abstract class

Vehicle class

{

no-of-types

void start( )

Car

↓

start, no-of-tyre

↓ ↓

key, 4

Bike

↓

start

↓

no-of-tyre

↓

2

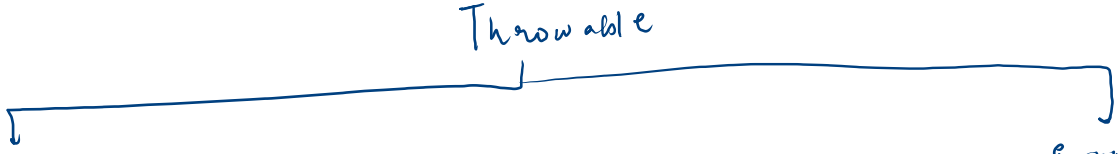
}

class car extends vehicle {

}

Note :- → we can't create instance of abstract class.

→ But, we can create the reference.



- Program.
- Recoverable
- 2 types
  - Checked Exception / Compile Time exception
  - Unchecked Exception / Runtime Exception

- lack of resources.
- Not easily recoverable
- Unchecked / R.T. Exception

Throwable

Exceptions

- Compile Time / Checked
  - → Class Not Found Exception
  - → IO Exception
    - ↳
    - ↳
  - → No Such Method Exception
  - → Sql Exception
  - →
- Runtime / Unchecked Exception
  - → Arithmetic Exception
  - → Null Pointer Exception
  - → Array Index Out of Bound
    - ↳ (5) → 0-4 arr[5]

Error

- ↳ VirtualMachine Error
  - ↳ StackOverflowError
- ↳ Linkage Error

Keywords use to handle exception

1. try
2. catch
3. finally
4. throw
5. throws

radius (r) → area = πr^2

↳ r = -ve → area = -ve ✗

Syntax

throw new Exception class ("message")

```java
class RadiusCannotBeNegativeException extends Exception{
    RadiusCannotBeNegativeException(String message){
        super(message);
    }
}
```

```java
public class index {

    Run | Debug
    public static void main(String []args) throws Exception {

            Scanner scn=new Scanner(System.in);
            System.out.println(x: "Enter the radius");
            int radius=scn.nextInt();

            if(radius<0){
                throw new RadiusCannotBeNegativeException(message: "You cann't enter negative radius");


            }else {
                System.out.println("Area is: "+ (radius*radius*3.14));
            }


    }

}
```

Class voting

↳ age , name, country

throw

↳ age < 18 → Exception

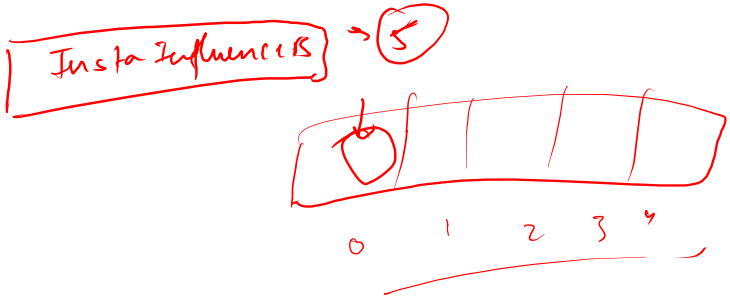
↳ country → India → Exception

print = "Yes, Candidate is eligible to vote"

```java
import java.util.*;

class AgeLimitException extends Exception{
    AgeLimitException(String message){
        super(message);
    }
}

class CitizenshipException extends Exception{
    CitizenshipException(String message){
        super(message);
    }
}

class Voting{
    int age;
    String name;
    String countryName;

}
```

```java
public class index {

    Run | Debug
    public static void main(String []args) throws Exception {
        Voting v=new Voting();

        v.age=34;
        v.name="Ram";
        v.countryName="India";

        if(v.age<18){
            throw new AgeLimitException(message: "You are under 18");
        }else if(!v.countryName.equals(anObject: "India")){
            throw new CitizenshipException(message: "You are not from India");
        }else {
            System.out.println(x: "Yes, Candidate is eligible to vote");
        }
    }

}
```

```java
class InstaIf{
    String name;
    int totalPosts;
    int totalReels;
    boolean blueTick;
    int followers;
    int following;
    String category;
    char gender;


    public InstaIf(String name,int totalPosts,int totalReels,boolean blueTick, int followers,int following, String
category,char gender){
        this.name=name;
        this.totalPosts=totalPosts;
        this.totalReels=totalReels;
        this.blueTick=blueTick;
        this.followers=followers;
        this.following=following;
        this.category=category;
        this.gender=gender;

    }
}
```




```java
public class Solution {

    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class s
        Scanner scn=new Scanner(System.in);
        int n=scn.nextInt();

        InstaIf arr[]=new InstaIf[n];

        for(int i=0;i<n;i++){
            String name=scn.next();
            int tP=scn.nextInt();
            int tR=scn.nextInt();
            boolean bT=scn.nextBoolean();
            int followers=scn.nextInt();
            int following=scn.nextInt();
            String cg=scn.next();
            char g=scn.next().charAt(0);

            InstaIf obj=new InstaIf(name,tP,tR,bT,followers,following,cg,g);

            arr[i]=obj;
        }

        for(int i=0;i<n;i++){
            InstaIf obj =arr[i];

            System.out.println(obj.name);
            System.out.println(obj.totalPosts);
            System.out.println(obj.totalReels);
            System.out.println(obj.blueTick);
            System.out.println(obj.followers);
            System.out.println(obj.following);
            System.out.println(obj.category);
            System.out.println(obj.gender);

        }
    }
}
```

```java
import java.io.*;
import java.util.*;

class YoutubeVideos{
    String song;
    String artist;
    int views;
    int likes;
    int releaseYear;
    int n;
    ArrayList<String>arr;

    public YoutubeVideos (String song,String artist,int views,int likes,int releaseYear,int n,ArrayList<String>arr){
        this.song=song;
        this.artist=artist;
        this.views=views;
        this.likes=likes;
        this.releaseYear=releaseYear;
        this.n=n;
        this.arr=arr;
    }
}
```

```java
public class Solution {

    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be named S
        Scanner scn=new Scanner(System.in);

        String song=scn.nextLine();
        String artist=scn.nextLine();
        int views=scn.nextInt();
        int likes=scn.nextInt();
        int releaseYear=scn.nextInt();
        int n=scn.nextInt();
        scn.nextLine();
        ArrayList<String>arr=new ArrayList<>();


        for(int i=0;i<n;i++){
            String s=scn.nextLine();
            arr.add(s);
        }


        YoutubeVideos obj=new YoutubeVideos(song,artist,views,likes,releaseYear,n,arr);

        System.out.println(obj.song);
        System.out.println(obj.artist);
        System.out.println(obj.views);
        System.out.println(obj.likes);
        System.out.println(obj.releaseYear);
        System.out.println(obj.n);

        arr=obj.arr;

        for(int i=0;i<arr.size();i++){
            System.out.println(arr.get(i));
        }


    }
}
```

```
class YoutubeVideos{
    String song;
    String artist;
    int views;
    int likes;
    int releaseYear;
    int n;
    ArrayList<String>arr;

    public YoutubeVideos (String song,String artist,int views,int likes,int releaseYear,int n,ArrayList<String>arr){
        this.song=song;
        this.artist=artist;
        this.views=views;
        this.likes=likes;
        this.releaseYear=releaseYear;
        this.n=n;
        this.arr=arr;
    }

    public void getProp(){

        System.out.println(song);
        System.out.println(artist);
        System.out.println(views);
        System.out.println(likes);
        System.out.println(releaseYear);
        System.out.println(n);


        for(int i=0;i<arr.size();i++){
            System.out.println(arr.get(i));
        }
    }
}
```

```
public class Solution {

    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class shou
        Scanner scn=new Scanner(System.in);

        String song=scn.nextLine();
        String artist=scn.nextLine();
        int views=scn.nextInt();
        int likes=scn.nextInt();
        int releaseYear=scn.nextInt();
        int n=scn.nextInt();
        scn.nextLine();
        ArrayList<String>arr=new ArrayList<>();


        for(int i=0;i<n;i++){
            String s=scn.nextLine();
            arr.add(s);
        }


        YoutubeVideos obj=new YoutubeVideos(song,artist,views,likes,releaseYear,n,arr);

        obj.getProp();



    }
```

```java
class YoutubeVideos{
    String song;
    String artist;
    int views;
    int likes;
    int releaseYear;
    int n;
    ArrayList<String>arr;

    // public YoutubeVideos (String song,String artist,int views,int likes,int releaseYear,int n,ArrayList<String>arr){
    //     this.song=song;
    //     this.artist=artist;
    //     this.views=views;
    //     this.likes=likes;
    //     this.releaseYear=releaseYear;
    //     this.n=n;
    //     this.arr=arr;
    // }

    public void setProp(String song,String artist,int views,int likes,int releaseYear,int n,ArrayList<String>arr){
        this.song=song;
        this.artist=artist;
        this.views=views;
        this.likes=likes;
        this.releaseYear=releaseYear;
        this.n=n;
        this.arr=arr;
    }

    public void getProp(){

        System.out.println(song);
        System.out.println(artist);
        System.out.println(views);
        System.out.println(likes);
        System.out.println(releaseYear);
        System.out.println(n);


        for(int i=0;i<arr.size();i++){
            System.out.println(arr.get(i));
        }
    }
}

public class Solution {

    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should
        Scanner scn=new Scanner(System.in);

        String song=scn.nextLine();
        String artist=scn.nextLine();
        int views=scn.nextInt();
        int likes=scn.nextInt();
        int releaseYear=scn.nextInt();
        int n=scn.nextInt();
        scn.nextLine();
        ArrayList<String>arr=new ArrayList<>();


        for(int i=0;i<n;i++){
            String s=scn.nextLine();
            arr.add(s);
        }


        YoutubeVideos obj=new YoutubeVideos();
        obj.setProp(song,artist,views,likes,releaseYear,n,arr);

        obj.getProp();
```

```java
import java.io.*;
import java.util.*;

class YoutubeVideos{
    String song;
    String artist;
    int views;
    int likes;
    int releaseYear;
    int n;
    ArrayList<String>arr;

    public YoutubeVideos (String song,String artist,int views,int likes){
        this.song=song;
        this.artist=artist;
        this.views=views;
        this.likes=likes;

        System.out.println("The song has just released");
    }

    public YoutubeVideos (String song,String artist,int views,int likes,int releaseYear){
        this.song=song;
        this.artist=artist;
        this.views=views;
        this.likes=likes;
        this.releaseYear=releaseYear;

        System.out.println("The song was released in " + releaseYear);



    }

}
```

```java
public class Solution {

    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class sh
        Scanner scn=new Scanner(System.in);

        String song=scn.nextLine();
        String artist=scn.nextLine();
        int views=scn.nextInt();
        int likes=scn.nextInt();
        int releaseYear=scn.nextInt();
        int n=scn.nextInt();
        scn.nextLine();
        ArrayList<String>arr=new ArrayList<>();


        for(int i=0;i<n;i++){
            String s=scn.nextLine();
            arr.add(s);
        }

        int k=scn.nextInt();

        if(k==1){
            YoutubeVideos obj=new YoutubeVideos(song,artist,views,likes);
        }
        else if(k==2){
            YoutubeVideos obj=new YoutubeVideos(song,artist,views,likes,releaseYear);
        }




    }
}
```

```java
class YoutubeVideos{
    String song;
    String artist;
    int views;
    int likes;
    int releaseYear;
    int n;
    ArrayList<String>arr;

    public YoutubeVideos (String song,String artist,int views,int likes,int releaseYear,int n,ArrayList<String>arr){
        this.song=song;
        this.artist=artist;
        this.views=views;
        this.likes=likes;
        this.releaseYear=releaseYear;
        this.n=n;
        this.arr=arr;
    }



    void add(int x){
        views+=x;
        System.out.println("Views are added");
        System.out.println(views);
        System.out.println(likes);
    }

    void add(int x,int y){
        views+=x;
        likes+=y;

        System.out.println("Views and likes are added");
        System.out.println(views);
        System.out.println(likes);
    }

}
```

```java
public class Solution {

    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be
        Scanner scn=new Scanner(System.in);

        String song=scn.nextLine();
        String artist=scn.nextLine();
        int views=scn.nextInt();
        int likes=scn.nextInt();
        int releaseYear=scn.nextInt();
        int n=scn.nextInt();
        scn.nextLine();
        ArrayList<String>arr=new ArrayList<>();

        if(n==5) n-=1;

        for(int i=0;i<n;i++){
            String s=scn.nextLine();
            arr.add(s);
        }

        YoutubeVideos obj = new YoutubeVideos(song,artist,views,likes,releaseYear,n,arr);

        int k=scn.nextInt();

        if(k==1){
            int x=scn.nextInt();
            obj.add(x);
        }
        else if(k==2){
            int x=scn.nextInt();
            int y=scn.nextInt();

            obj.add(x,y);
        }




    }
}
```

```
import java.io.*;
import java.util.*;

class YoutubeVideos{
    String song;
    String artist;
    int views;
    int likes;
    int releaseYear;
    int n;
    ArrayList<String>arr;
    static int totalVideos=0;

    public YoutubeVideos (String song,String artist,int views,int likes,int releaseYear,ArrayList<String>arr){
        this.song=song;
        this.artist=artist;
        this.views=views;
        this.likes=likes;
        this.releaseYear=releaseYear;
        this.arr=arr;
    }


}
```

```
public class Solution {

    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be
        Scanner scn=new Scanner(System.in);

        int x=scn.nextInt();
        scn.nextLine();
        for(int i=0;i<x;i++){
            String song=scn.nextLine();
            String artist=scn.nextLine();
            int views=scn.nextInt();
            scn.nextLine();
            int likes=scn.nextInt();
            scn.nextLine();
            int releaseYear=scn.nextInt();
            int n=scn.nextInt();
            scn.nextLine();
            ArrayList<String>arr=new ArrayList<>();

            for(int j=0;j<n;j++){
                String s=scn.nextLine();
                arr.add(s);
            }

            YoutubeVideos obj = new YoutubeVideos(song,artist,views,likes,releaseYear,arr);
            obj.totalVideos+=1;
        }

        System.out.println(YoutubeVideos.totalVideos);

    }
}
```

(1) Syntax

```
try {
    // risky code
}
catch ( Exception Class Name  ref var-name )
{
    // handling code.
}
```

final int a = 10;

a += 20;