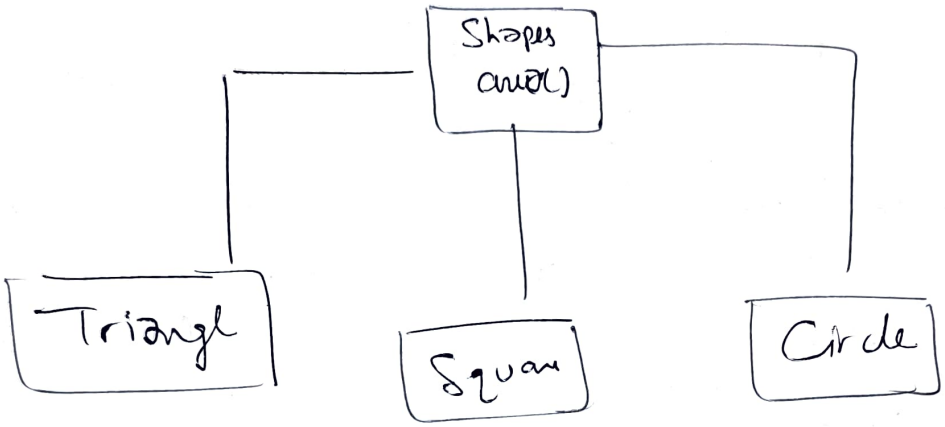
Polymorphism

Many form → kisi object ka

Many ways to represent

Language
- Not support — Object based language.
- support — object oriented lang.

- Occur during inheritance.



Types of Polymorphism

(1) Static / Compile Time

Achieved by method overloading.

Class has multiple methods with same name

Same
- Name

diff
- type, argument, ordering, number

Ex: Multiple Constructors

```
A a = new A();
A a = new A(3,4);
```

Java determine which method will be called at compile time.

2 Dynamic / Runtime Polymorphism

Achieved by method Overriding.

Parent
M1()
S

Child
M2()
S

if M1 = M2

@Override // Annotation (Access) (Run)

Parent obj = new Child();

Here which method will be called depend on child

This is known as Upcasting.

How Java determine this? → Dynamic Method Dispatch

- You can't override final Method
- can you override static Method ✗ Not applied on instance variable

Encapsulation

Wrapping up the implementation of the data members & methods in a class.

Abstraction

Hiding unnecessary details & showing valuable information

only key is needed

Encapsulation v/s Abstraction

principle ← How internally made don't care

Also known as abstract data type (ADT)

ArrayList

Abstraction solving design level issue Encapsulation

Solution - implementation level

OOPs - 4

```
Public class A {
    int num;
    String name;
    int [] arr;

    Public A( int num, Str ---- ,---- ){
        this.num = num;
        this.name = name;
        this.arr = new int [num];
    }
```

Main fn

```
psvm {
    A obj = new A( 34, ALL);
    // Need to do few thing
```

① Access the data members

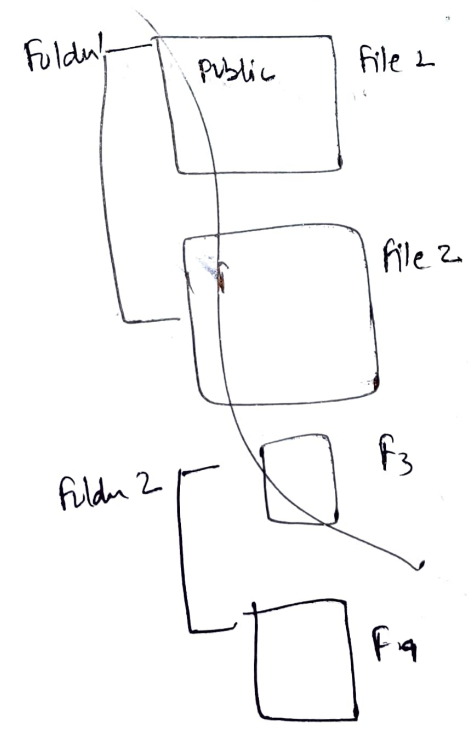
② Modify " " "

{ 700 line of code }

```
ArrayList <Integer> = new ALL();
    list.DEFAULT CAPACITY
```

← can't access

Access Modifier



Access Modifier	Access
Public	F1 + F2 + All + F3
Private	F1 ✓ F2 ✗
default	F1 + F2 + Folder2

	Class	Package	Subclass	Subclass (diff pkg)	World
Public	✓	✓	✓	✓	✓
Protected	✓	✓	✓	✗	✗
No modifier	✓	✓	✗	✗	✗
Private	✓	✗	✗	✗	✗

- Public → Everywhere
- Private → Only in same class
- Protected → Inheritance
- default → Same Pkg + Class

Java Packages

- Build in Packages
  - Package consist large no of class which are a part of Java API
  - eg: java.util, java.io, java.applet
- User Defined Packages
  - encapsulate classes, sub-classes & interfaces
  - OOPS — Lec4 [1st.java, 2nd.java]
  - Package OOPs.Lec4;
  - to import a class from package
  - import OOPs.Lec4.MyClass

Name Conflicts

Pack java.util → Date
Pack java.sql → Date

import java.util.Date
// or java.sql.*;

Accessing of Access Modifiers

	Same Class	Same Pkg	Subclass (Diff Pkg)	Non-Sub (Diff Pkg)
1. Public → All	✓	✓	✓	✓
2. Protected → Till inheritance	✓	✓	via inheritance (✓)	✗
3. default	✓	✓	✗	✗
4. Private	✓	✗	✗	✗

Java Universities

Packages In-built

- lang → Automatically Imported
- io → File reading → System.in, out, Stream
- util → data Structure & Coll framework → LL, Bit, etc
- applet → to create applets → Java pgm that run inside web browser. → Make websites
- For GUI → awt
- net → networking Lang

Object is a class

↑ Root of Inheritance hierarchy

By default extends Object Class

Public class SubClass extends A

↳ extend A + extend object class

Internally

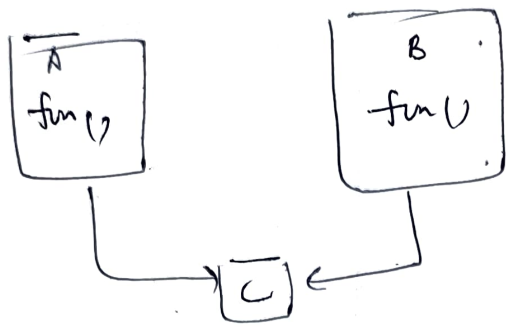
hashcode
↳ Random value not address
↳ To give unique representation to object

== V/S .equal

- == → Comparator → Point to same obj or not
- .equal → Method Content Checking.

OOP-5

ABSTRACT CLASSES & INTERFACES



c.fun() get confuse

Multiple Inheritance ✓

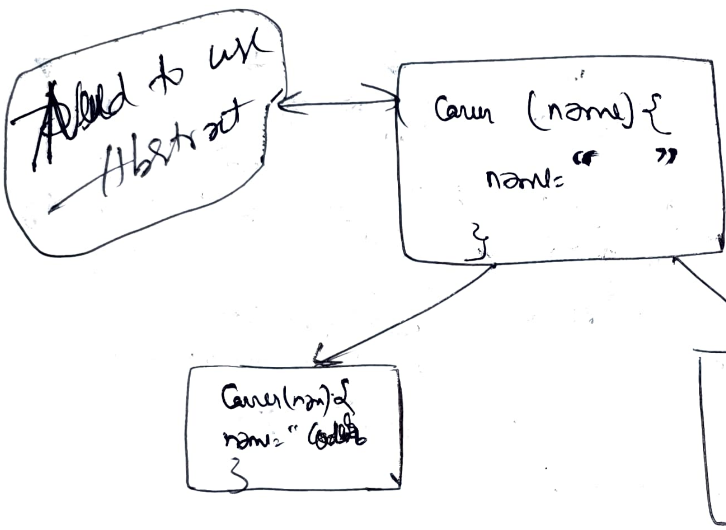
abstract class
- abstract method ✓
- abstract method

Java not support

abstract void make sound();

Every class that contains must implement this method

Only method signature, no body ✓



Defining body from itself.

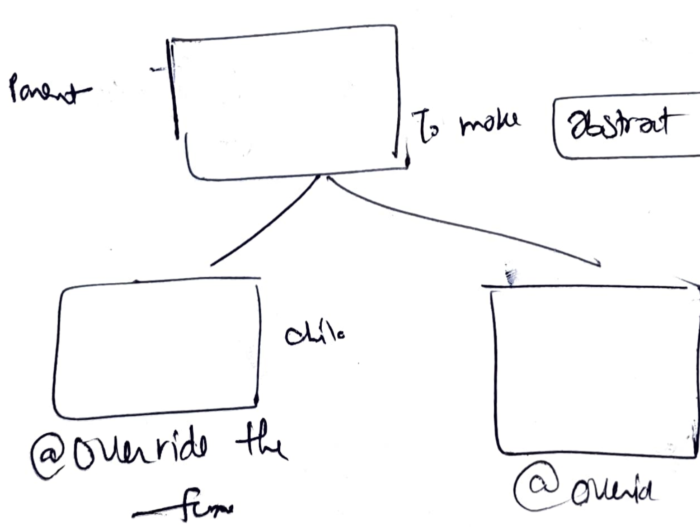
@override Annotation

Improve code readibility

Not use it will be treated as new method

abstract void Career (String name);

If class contain abstract method should be contain abstract



To make abstract fun → make abstract class

abstract void Career();

@override

@override void Career() {

}

Can make constructor of abstract class ✓

Imp — Can you create object of abstract class ✗ — Can use as ref. variable

" " " abstract constructor ✗

" " " abstract static methods ✗

Needs to be @ override — Can't be overridden

Can we have final abstract class ✗

Can't inheritance bc final

by default: Public static & final

INTERFACE

abstract can provides implementation

A class can implement more than one interfaces.

Multiple inheritance is allowed.

Brake

```
break()
start() ✗
```

Engine

```
start(),
stop(),
Accelerate()
```

~~Engine class~~ Multi player

```
Mp player
start();
stop();
```

```
public interface Engine {
    void start();
    static final int price = 78k;
    void stop();
    void accelerate();
}
```

implements keyword

public class Car implements Engine, Brake

Interface

Default Variable are → Public, Static & final

→ use to achieve abstraction & multiple inheri.

ANNOTATIONS

	Class	~~Object~~ Interface
Instantiation	In class you can create objec	Can't ~~create~~ objec
Inheritance	Single	Multiple
Constructor	✓	✗
Access Modifier	P, P, P, d	By default → public

- Interface is Idea for achieving abstraction & multi. inheritance.