object oriented programming

Classes
↓
group of these
entities

object
↓
entities in the real world
↓
Ex: animal, watch, pen - property
↓
blue,
yellow,

change color()
↳ fun

class
pen

attributes
(properties)
+
functions
(behaviors)

color (string)
↑
chane color()

Getters & Setters

Get: to return the value

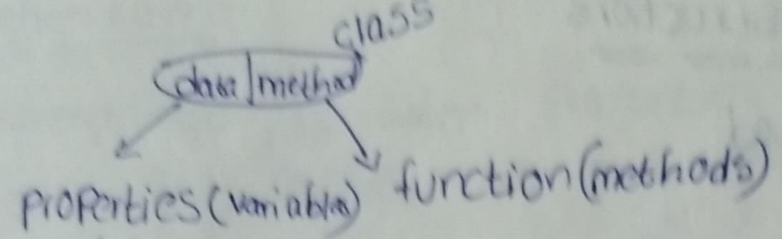
Set: to Modify the value

this: this keyword is used to refer to the current object

Encapsulation

Encapsulation is the one-of the core topic/concept in oops

→ it means hiding the internal details



final dat

single entity wrapping data

→ It improves data security

Encapsulation is the process wrapping data and functions into a single unit

Constructors

→ constructor is the special method ~~which is~~ ~~automatically~~ in ~~the~~ a class that is automatically called when an object of that class is created

key points

1) Constructors have the same name as class

2) Constructors don't have a return type (Not even void)

3) Constructors are only called once, at object creation

Types of Constructors

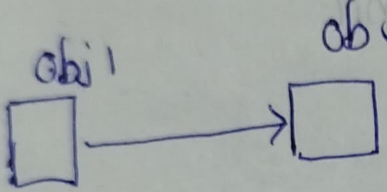
Non- Parameterized

Parameterized

Copy Constructor

Copy Constructors

obj 2 = copy (obj 1)

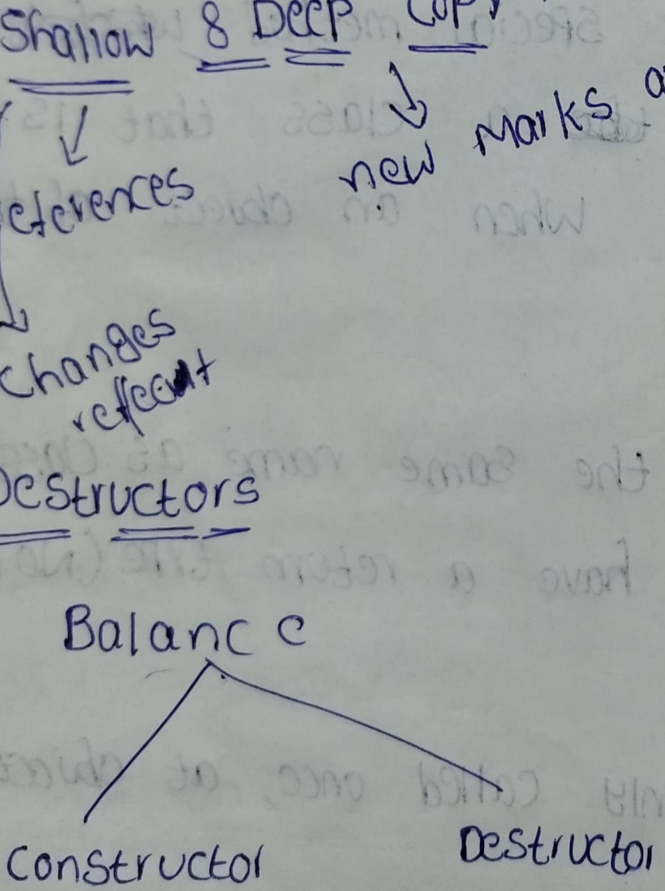


Shallow & Deep Copy → changes don't refect

Shallow ↓ references → changes refect

Deep Copy ↓ new Marks array

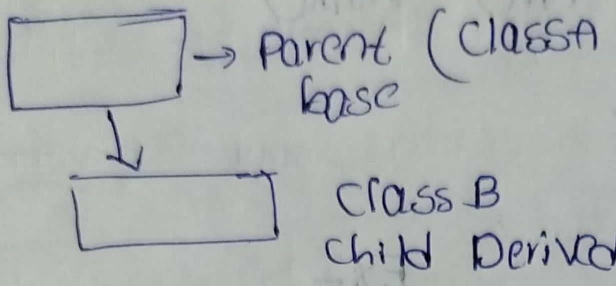
Destructors



Garbage Collector

Inheritance

Inheritance is when Properties & Methods of base class are passed on to a derived class



Types of Inheritance

1) Single level Inheritance

Base Class

↓

Derived class

2) Multi level Inheritance

Base Class

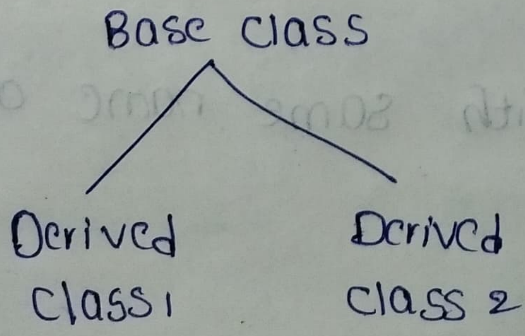
↓

Derived Class

↓

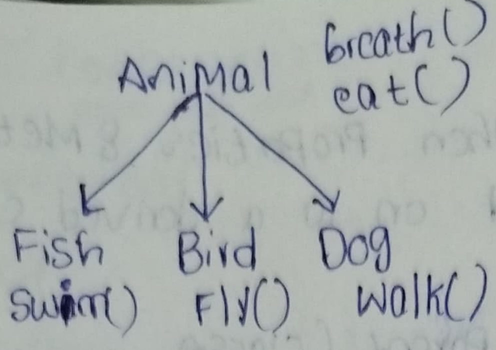
Derived Class

3) Hierarchial inheritance

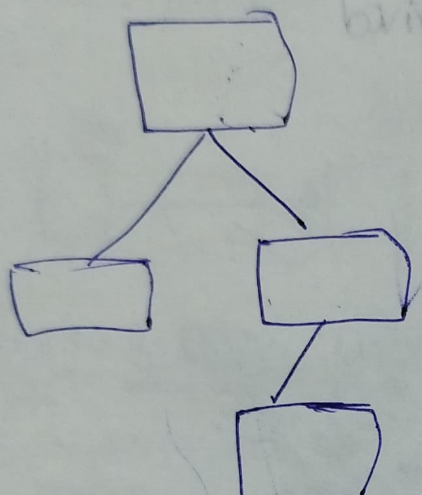


Note: one base class with different derived classes.

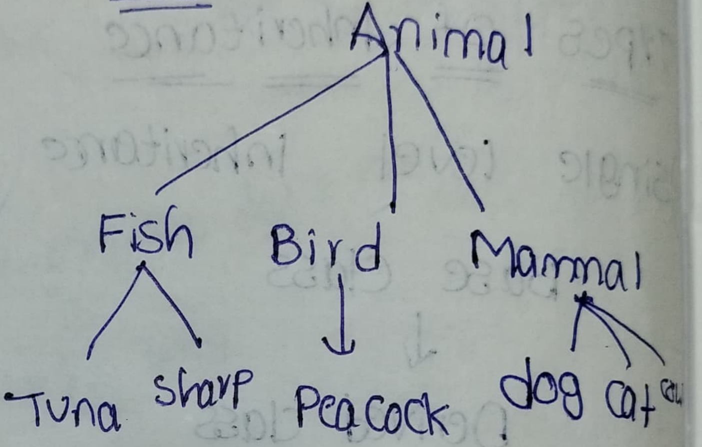
Ex:-



4) Hybrid Inheritance



Ex:-



Polymorphism

- Compile Time Polymorphism (static)
  • Method overloading
- Run Time Polymorphism (dynamic
  • Method overriding

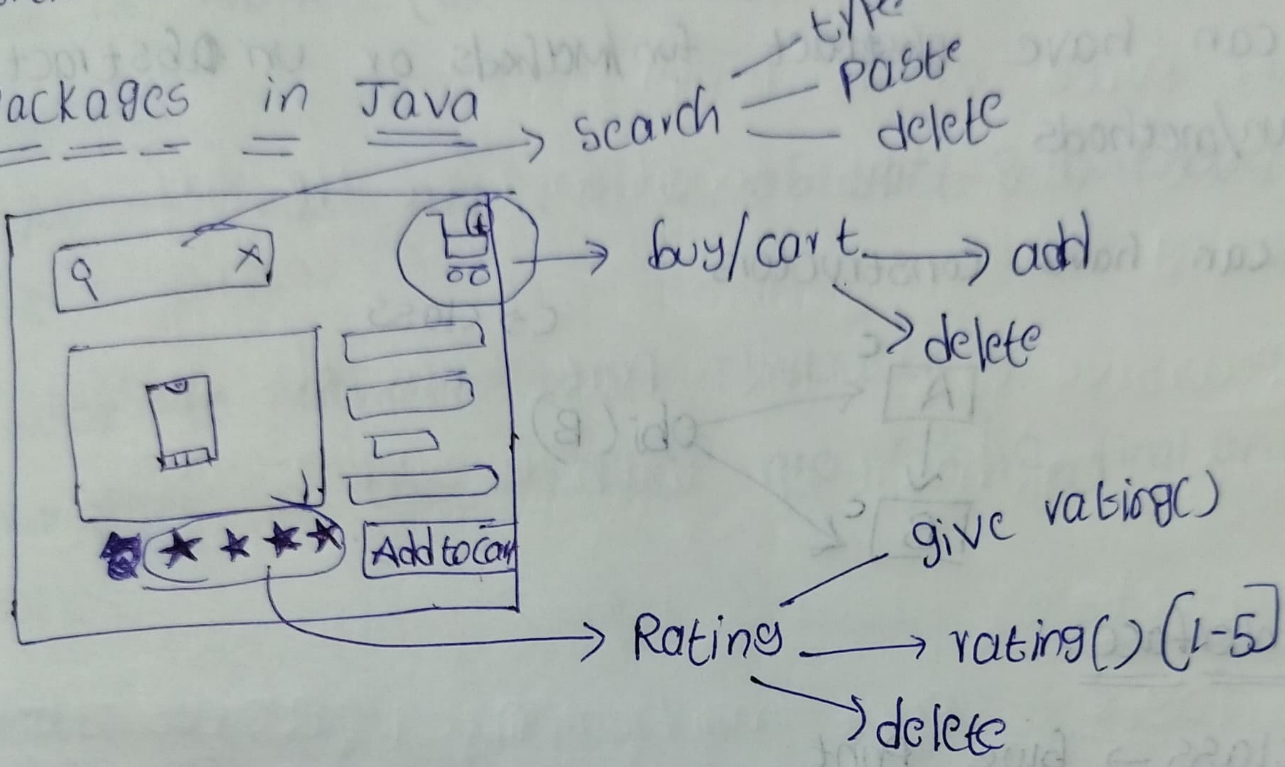
Method overloading

Multiple functions with same name and different Parameters.

Method overriding

parent and child classes both have same function with different defination

Packages in Java



-> Packages group of similar types of classes and interfaces and sub-packages

Packages are two- types

1) in built 2) user- defined

Abstraction

Hiding all unnecessary detail's and showing only important details/Parts to the user

Abstract clas   interfaces

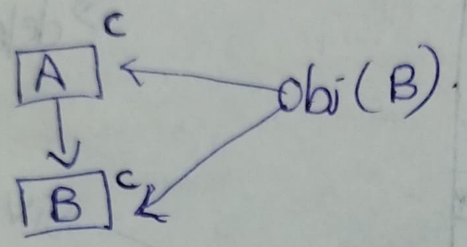
Abstract class

→ Can not create instance/object of abstract c

→ can have abstract fun/methods or un abstract fun/methods

→ can have constructors

C→ class



Interfaces

class → blue print

interface → blue print of class

→ properties

Ex: Car [wheels, speed, engine] (interface)

↓

Maruti 800 (class)

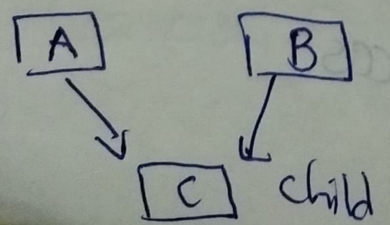
Car 1 M800

Car 2 M800

Care 3 M 800

(obj's)

Multiple Inheritance (5th)



- total abstraction (Interfaces)

Note÷ in abstrac class (0 - 100% range) → abstraction
but in interface (100%) abstraction

Properties

-> All Methods are Public, abstract & without implementation

-> used to achieve total abstraction

-> variables in the interface are Public, final and static

Static keyword

Static keyword in java is used to share same variable or method of a given class

- Properties
- Functions
- Blocks
- Nested Classes

Super keyword

Super keyword is used to refer immediate Parent class object.

-> To access Parent Properities
-> To acess Parent Methods/fun
-> To acess Parent constructor