Object Oriented Programming

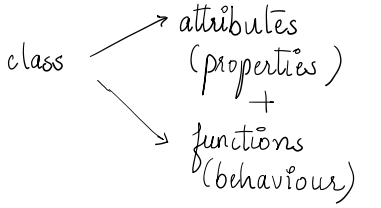
02 March 2023 18:34

Classes & Objects :

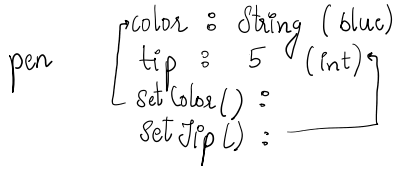
Classes ↓ group of these entities

Objects ↓ entities in the real world

Class : blueprint of an object



Eg : Let the object be pen



Access Modifiers :

Access Modifier	within class	within package	outside package by subclass only	outside package
Private	Y	N	N	N
Default	Y	Y	N	N
Protected	Y	Y	Y	N
Public	Y	Y	Y	Y

Getters & Setters :

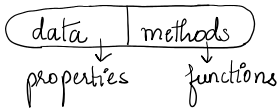
Get : to return the value

Set : to modify the value

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

Encapsulation :

Encapsulation is defined as the wrapping up of data & methods under a single unit. It also implements data hiding.



Constructors :

Constructor is a special method which is invoked automatically at the time of object creation.

- Constructors have the same name as class or structure.

- Constructors don't have a return type. (Not even void)
- Constructors are only called once, at object creation.
- Memory allocation happens when constructor is called.

Types of Constructor :

(i) Non-parameterized

(ii) Parameterized

(iii) Copy Constructor :

A copy constructor is made and the values are copied from the previous constructor.

Shallow & Deep Copy :

Shallow copy are those copy in which the references are copied. The changes are reflected.

In deep copy, a new array is made and the changes are not reflected.

Destructors :

We do not have destructors in Java instead we use garbage collector

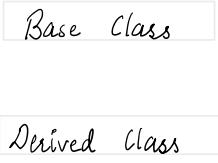
The garbage collector removes the object in the code which is not in use

Inheritance :

Inheritance is when the properties & methods of base class are passed on to derived class.

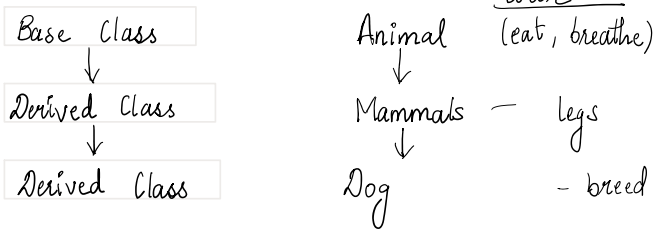
Types of Inheritance :

(i) Single Level Inheritance :

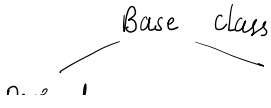


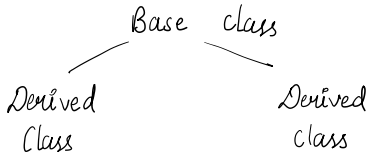
Example : fish class inherit from animal

(ii) Multi-Level Inheritance :



Hierarchical Inheritance :

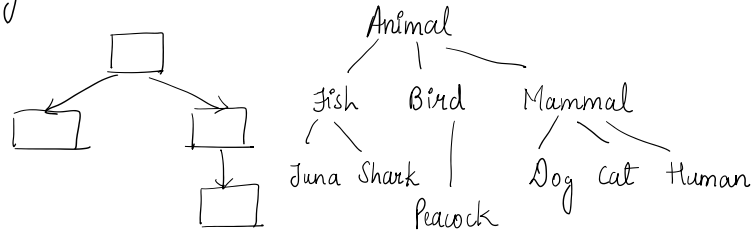




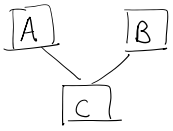
Eg : Animal → eat, breath

Bird Mammal
swim fly walk

Hybrid Inheritance



Multiple Inheritance :
Not in Java, but can be implemented using Inheritance.



Polymorphism :

Poly - many morph - forms.

Same function with different parameters and same name.

(i) Compile - time Polymorphism : (static)
- Method Overloading

(ii) Run - time Polymorphism :
- Method Overriding

Method Overloading :
Multiple functions with the same name but different parameters

```
Eg : Calculator {
        sum ( int a, int b)
        sum ( float a, float b)
        sum ( int a, int b, int c)
     }
```

Method Overriding :

Parent and child class both contain the same function with a different definition.

Eg : Animal
void eat (x) → "eat anything"

Deer
void eat(x) → "eats grass"

Packages in Java :

Packages is a group of similar types of class, interface and sub-classes..

(i) In-built Packages
(ii) User-defined Packages

Abstraction :

Hiding all the unnecessary details and showing only the important parts to the user.

Abstract Classes
Interfaces

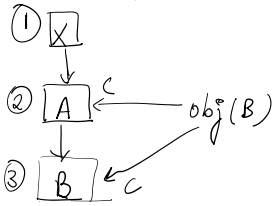
Abstract Classes :

(i) Cannot create an instance (object) of abstract class

(ii) Can have abstraction / non-abstract methods

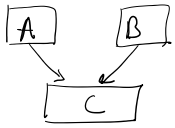
(iii) Can have constructors

While calling of constructor



Interfaces :

It is a blueprint of a class.



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

Maruti 800 (class)

car1 car2 car3 car4

- used to implement multiple inheritance
- to achieve total abstraction.

Interface	class
inherit	inherit
implement	extends

- All the methods are public, abstract & without implementation.
- used to achieve total abstraction
- Variables in the interface are final, public and static

Static Keyword :

Static keyword in Java is used to share the same variable or method of a given class.

- Properties
- Functions

Blocks

- Nested Classes

Super Keyword :

It is used to refer immediate parent class object.

- to access parent's properties
- to access parent's functions
- to access parent's constructor

NOTE : If a child object is made then it can be assigned to the parent reference variable.