Object Oriented Programming....

java is OOPL.

features of OOP is ....

1) Encapsulation

2) Inheritance

3) Abstraction

4) Polymorphism

These features neither belongs to java nor to any other language.

These are global features.

Any language can adopt these features to become OOPL.

Examples.....

C++ , java,.net, PHP........

In any language, concept of OOP feature is same, but implementation is according to syntax of language..

for example...

I want to print a msg in different languages....

C ----> printf("Hello");

C++ ---> cout>>"Hello";

Java ---> System.out.println("Hello");

.net ----> Console.write("Hello");

Note : In any language, we can implement these features using 2 things...

1) Object

2) class

Object : A real world entity/substance is called Object

Object is having 3 things....

1) Identity / name

2) Properties / variables

3) functionality / methods

Example :

Object type : Human

Identity : Syam

Properties : Color, Height, Weight, Age, Qualification......

Functionality : read( ) , see() , walk( )

swim( ), drive( ), teach( )......

Object type : Mobile

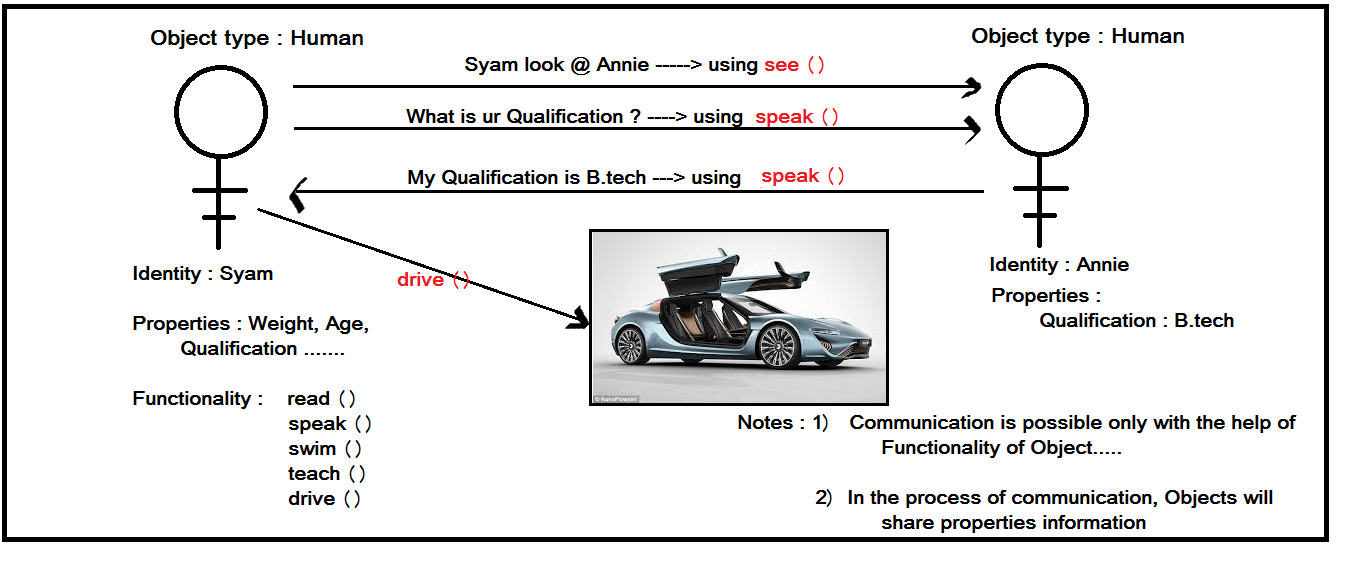
Identity : Samsung

Properties : Model, price, configuration, color......

Functionality : call( ), msg( ), browse(), play( )......

Note : Objects will participate in the process of communication using functionality(methods).

In the process of communication, Objects will share properties information.....



class :

Complete representation of real world entity(object).

class is a plan/model/blue print of an object.

using single model, we can construct "n" number of objects of same type.



Encapsulation :

Concept coming from "capsule".

what ever the content which is inside the capsule is sensitive, hence protected with cap.

We need to protect complete information of object......

Encapsulation is the concept of protecting object information by placing properties and funcationlity as a block...

class / Encapsulation

{

properties

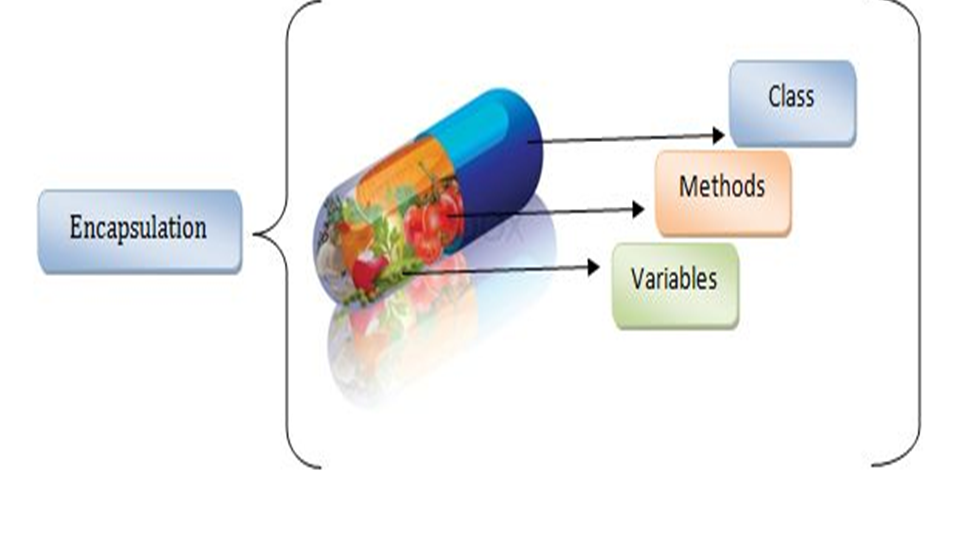
+

methods

}

Encapsulation ---> therory...

Class ---> technical implementation.....



Inheritance :

Creating new Object with the help of existing object.

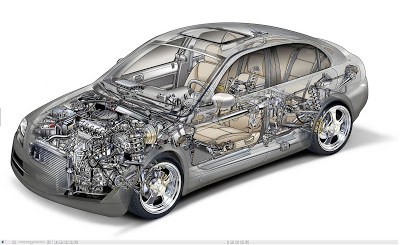
Releasing next version of existing object.....

Existing system ---> proposed system....



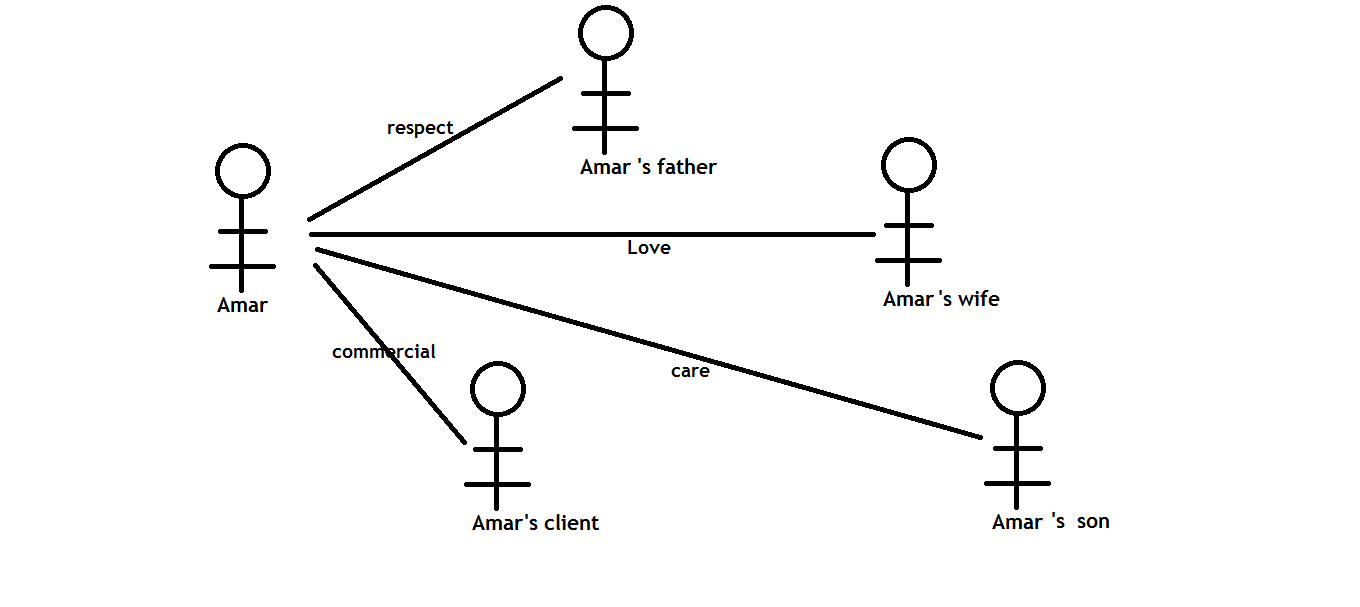
Abstraction :

Hiding un necessary details of Object and shows only essential features to communicate.



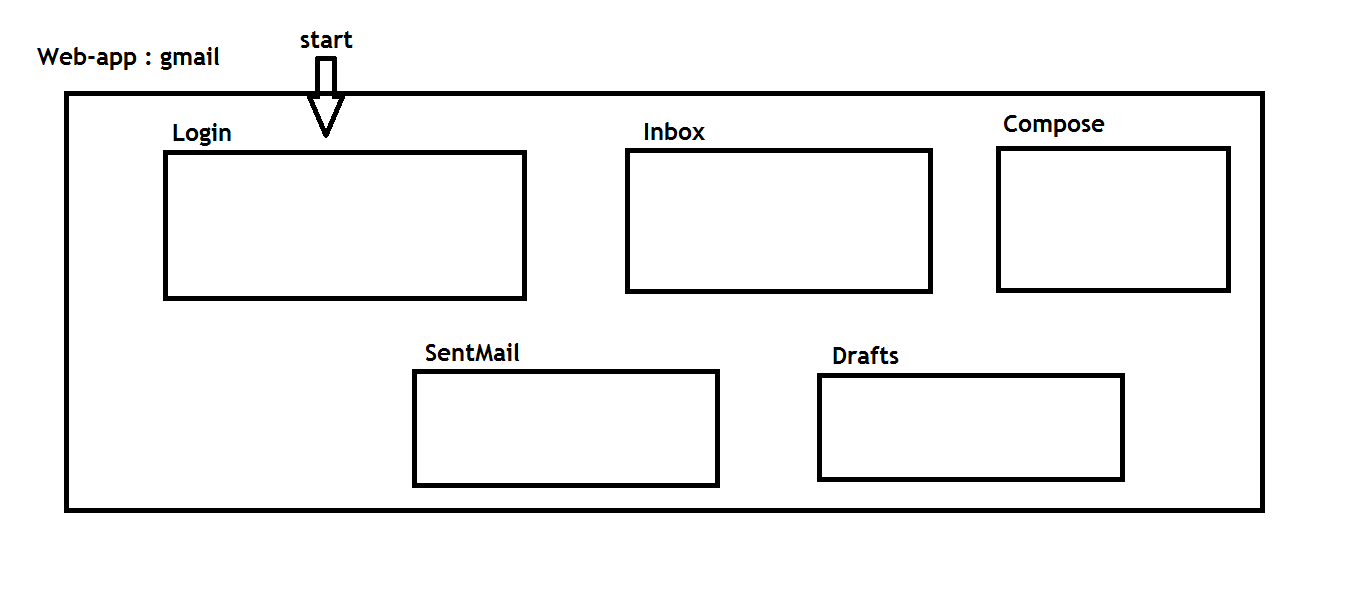
Polymorphism :

One Object is showing different behavior while communicating with different objects.



Structure of Java application:

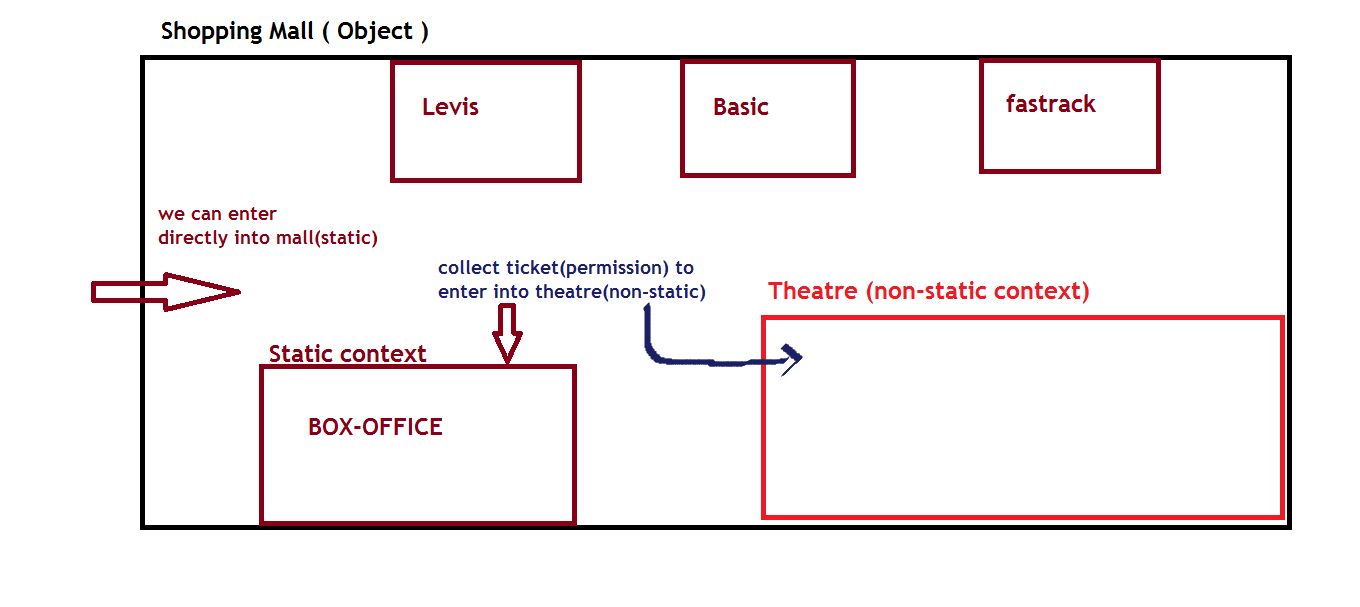
1. Every application is a set of files/programs.
2. Every java application is a set of class files.
3. Every class file is a Model that represents Object.
4. JVM will run java application.



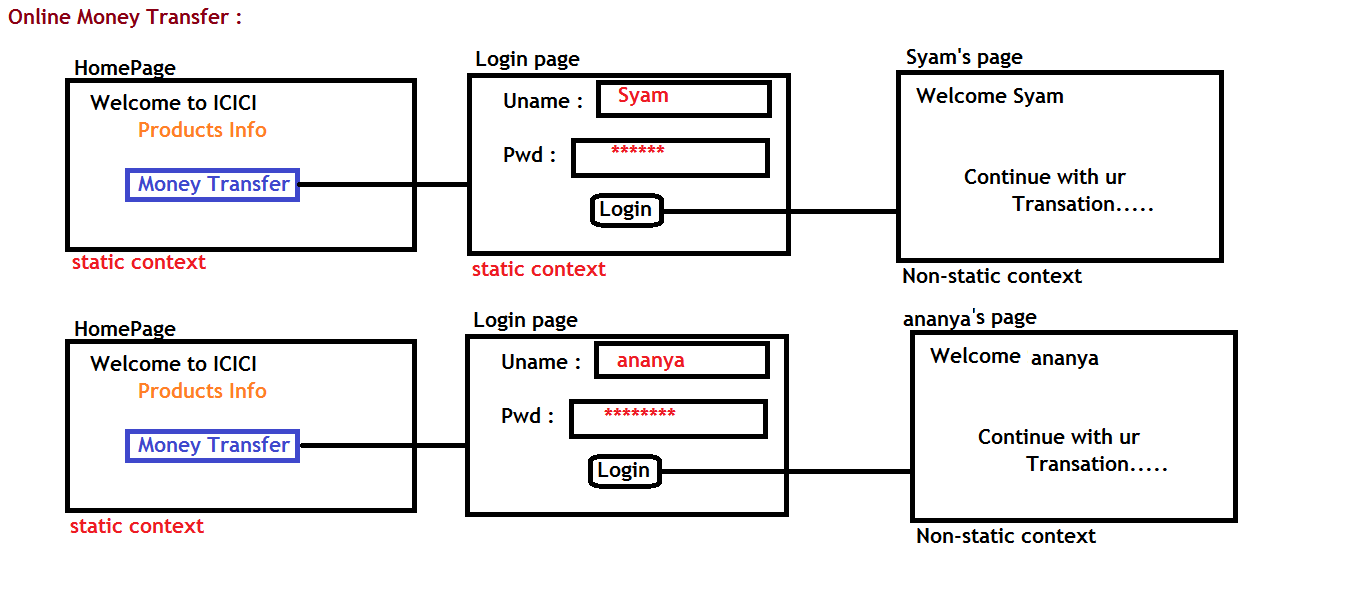
Every java application is mainly having 2 contexts:

1. Static context (free accessible area)
2. Non-static context ( restricted access/permissions required)

General example :



Technical example:



Class members :

1. Objects can be represented by classes only.
2. We need to write one class to define object.
3. Every instruction in java application must be placed inside the class.
4. All these following members allowed into a class.

