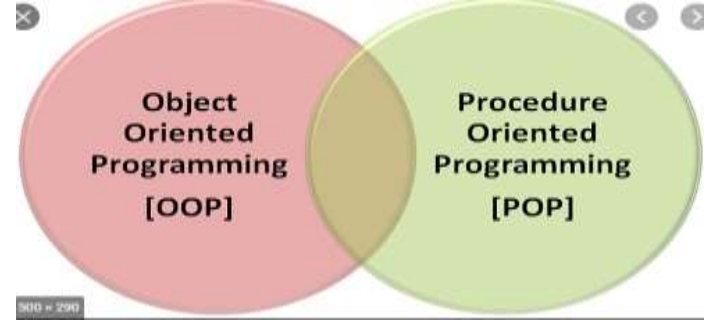


Object Oriented Programming - Java

OOPS and POP



OOP	POP
Object oriented.	Structure oriented.
Program is divided into objects.	Program is divided into functions.
Bottom-up approach.	Top-down approach.
Inheritance property is used.	Inheritance is not allowed.
It uses access specifier.	It doesn't use access specifier.
Encapsulation is used to hide the data.	No data hiding.
Concept of virtual function.	No virtual function.
C++, Java.	C, Pascal.

C PROGRAM

```
#include<stdio.h>
void main()
{
int a,b,c;
a=10,b=30;
c=a+b;
printf(“%d”,c);
}
```

What is java?

- Developed by Sun Microsystems (James Gosling)
- A general-purpose object-oriented language
- Based on C/C++

Introduction to JAVA

- JAVA was developed by Sun Microsystems Inc in 1991, later acquired by Oracle Corporation.
- It was developed by James Gosling.
- It is a simple programming language. Writing, compiling and debugging a program is easy in java. It helps to create modular programs and reusable code.

Applications

- Desktop Applications such as acrobat reader, media player, antivirus, etc.
- Games Designing
- Enterprise Applications such as banking applications.
- Mobile
- Embedded System
- Smart Card
- Robotics

OOPS Concept

- Object
- Class
- Inheritance
- Polymorphism
- Abstraction
- Encapsulation
- Data Hiding.

Java is an object-oriented programming language.
Everything in Java is an object.

Object

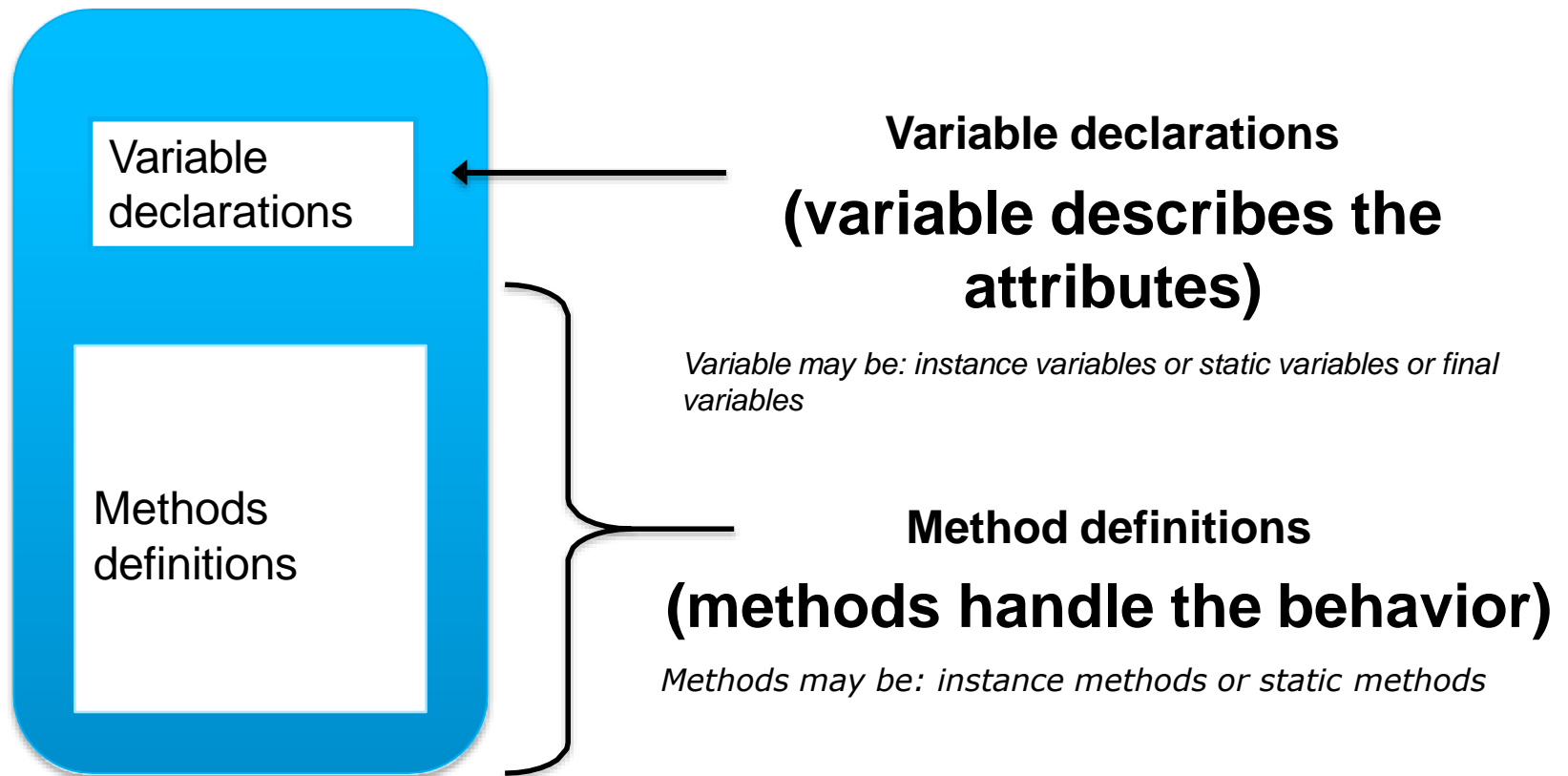
- Object is a **real time or run time entity that has state and behaviour.**
- **Instance of class.**
- That represents - a person , chair, pen, table, keyboard, bike, etc. (It can be physical or logical).
- Eg:
Human → State (name, colour, type, Height)
→ Behaviour (run(), eat(), read(), write())

class

- Class is **user defined data type**.
- Collection of objects is called class. It is a **logical entity**.
- **Blueprint** that object follows.
- A class contains **variable declarations** and **method definitions**.
- Classes can not communicate with each other.
 - Need to create object to access data member (state) and member function (behavior) of other class.

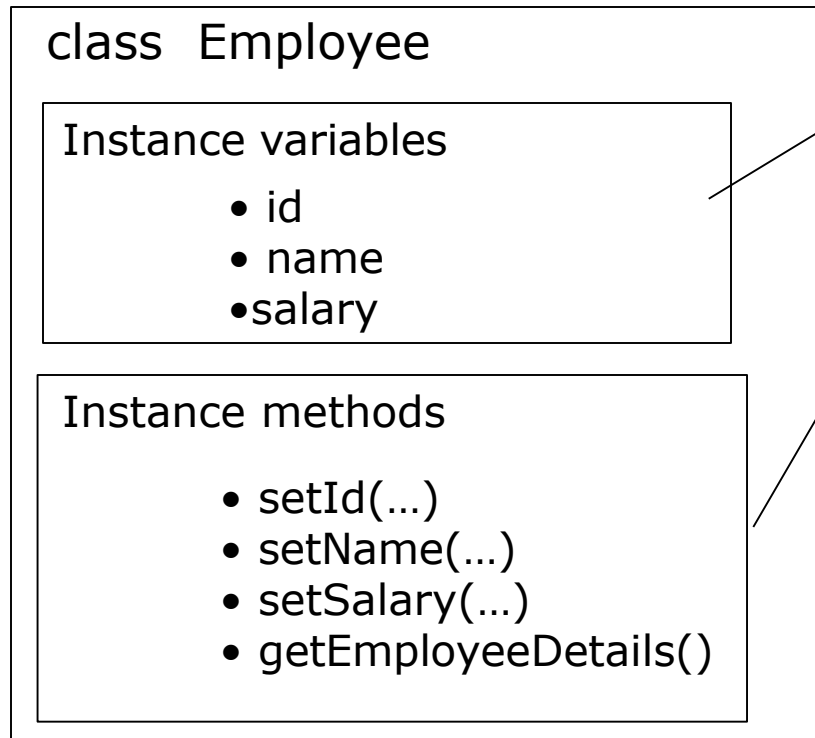
Classes

A class contains **variable declarations and method definitions**



Defining a Class in java

Define an Employee class with instance variables and instance methods



class Employee{

```
int id;
String name;
int salary;
```

```
void setId(int i) {
    id = i;
}
void setName(String n) {
    name = n;
}
void setSalary(int s) {
    salary = s;
}
void getEmployeeDetails( ) {
    System.out.println (name + " salary is " + salary);
}
```

}

Abstraction

- Showing / representing **only essential features** without including the background detail or explanation.
- Hiding internal details and showing essential functionality is known as abstraction.
- **Look essential and ignore the rest.**
- Eg: ATM, CAR, App.

By using Abstraction we will achieve

- Security
- Enhancement is very easy
- Maintainability

Encapsulation

- Binding Variables and methods into a single unit or entity.
- Every java class is a example for abstraction.



Capsule

- Encapsulation =Data hiding + abstraction

Example

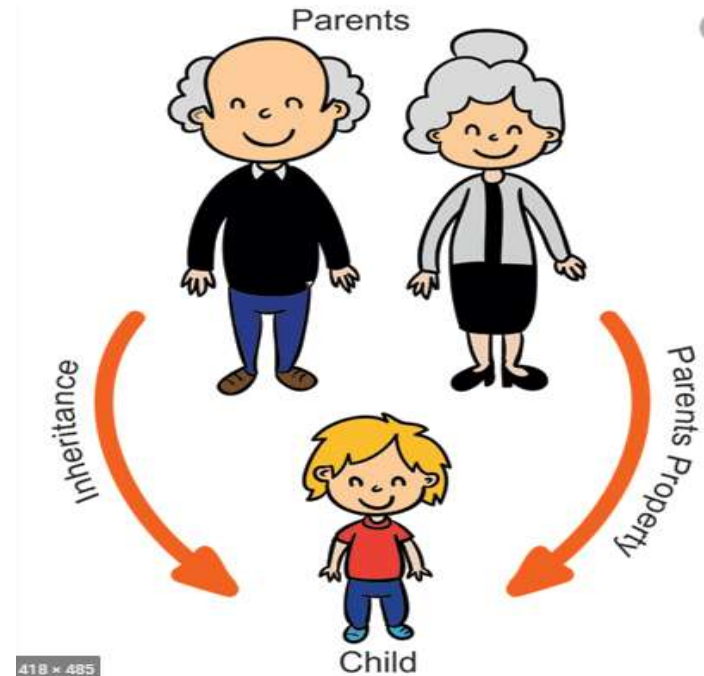
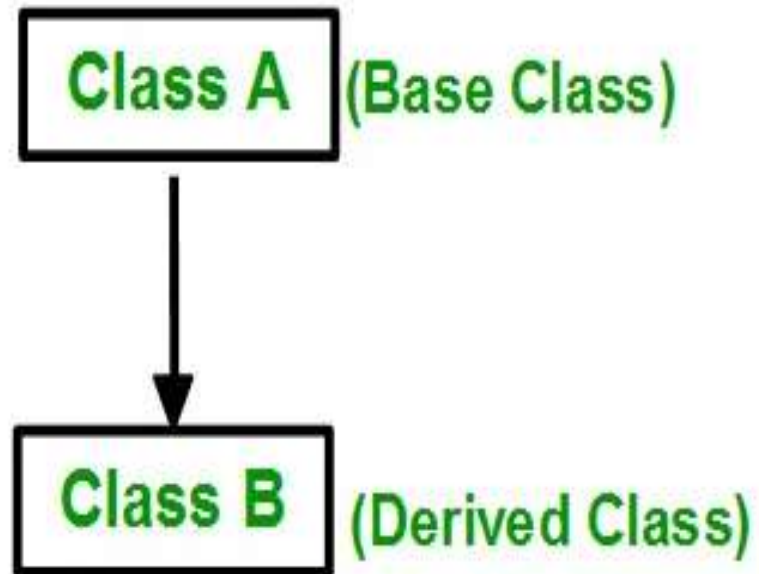
Class Student

```
{    String name;  
    int mark1;  
    int mark2;  
    Void add()  
    {  
        System.out.println(mark1+mark2);  
    }  
}
```

Inheritance

- **Inheritance** is a mechanism in which **one class acquires the property of another class**
- *When one object acquires all the properties and behaviours of a parent object, it is known as inheritance.*
- It provides **code reusability**.
- It is used to achieve **runtime polymorphism**

Example



- **Super Class:**

The class whose features are inherited is known as superclass(or a base class or a parent class).

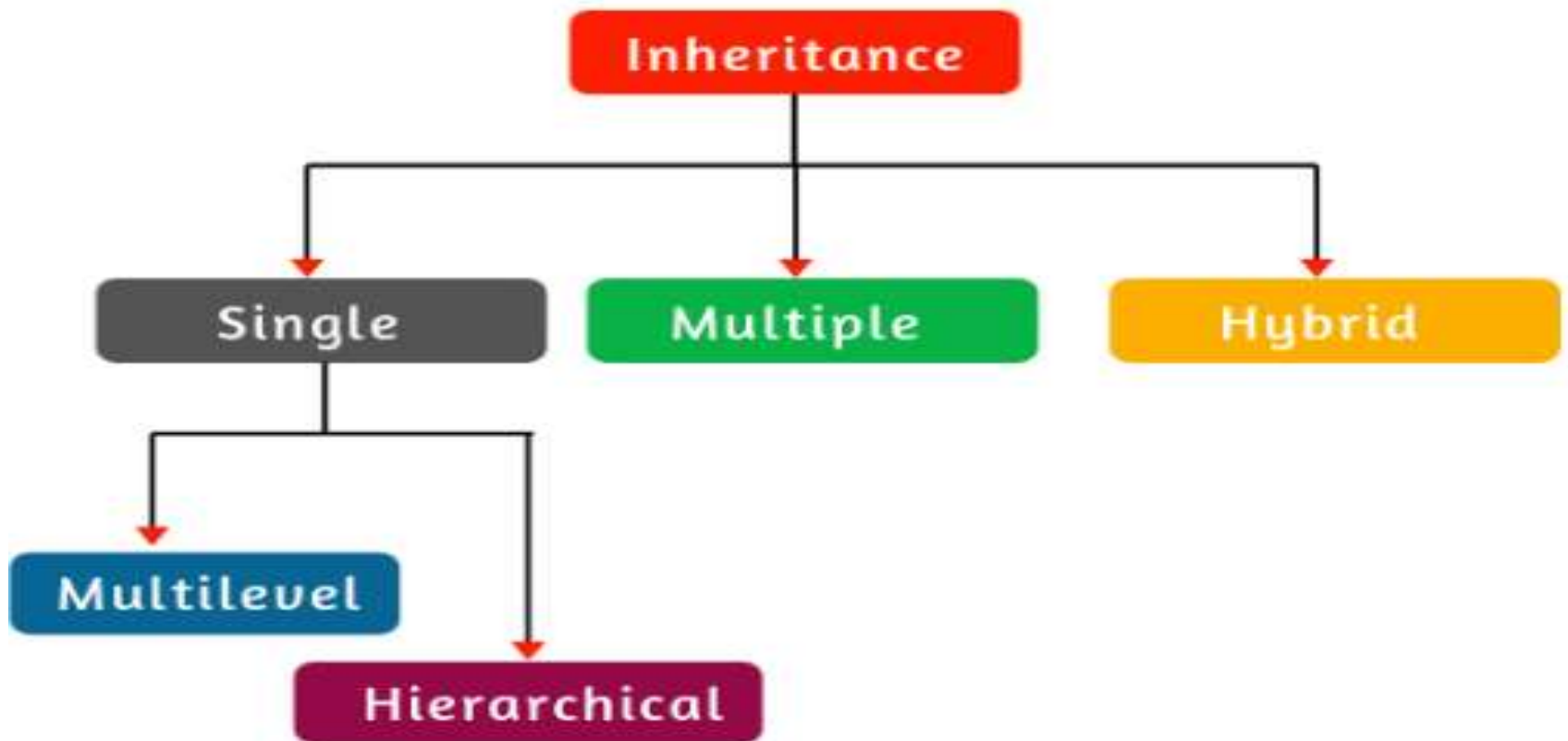
- **Sub Class:**

The class that inherits the other class is known as subclass(or a derived class, extended class, or child class). The subclass can add its own fields and methods in addition to the superclass fields and methods.

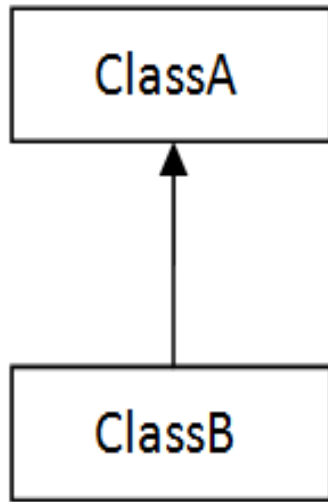
Who is giving – super, base or a parent class

who is acquiring - Sub Class, derived class, extended class, or child class.

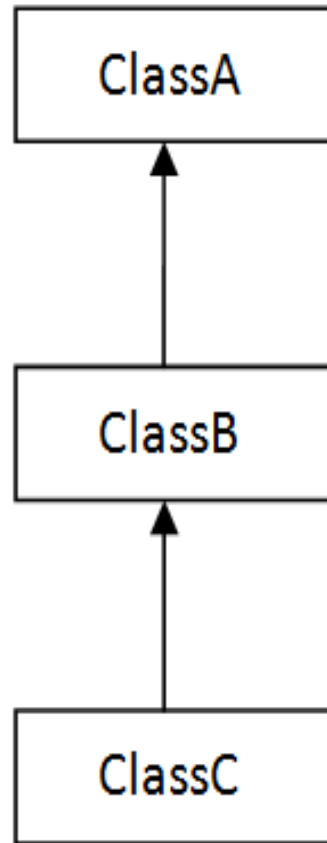
Types



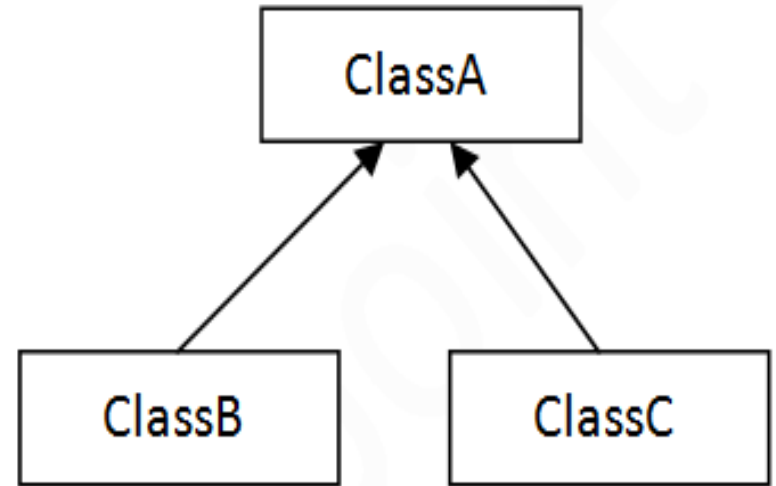
Types of inheritance in java



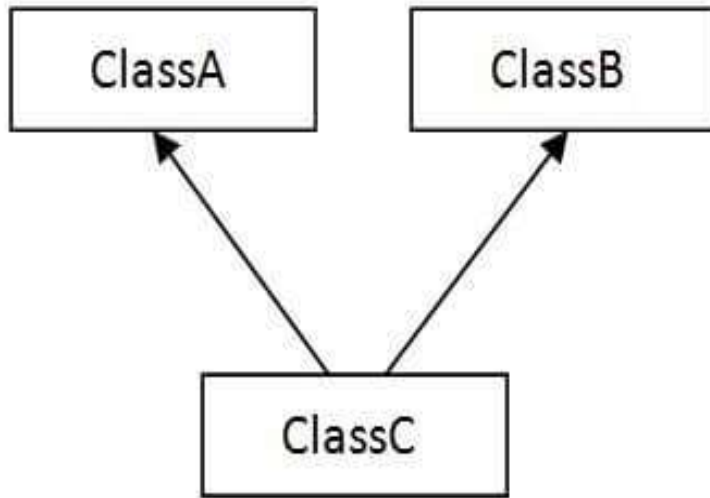
1) Single



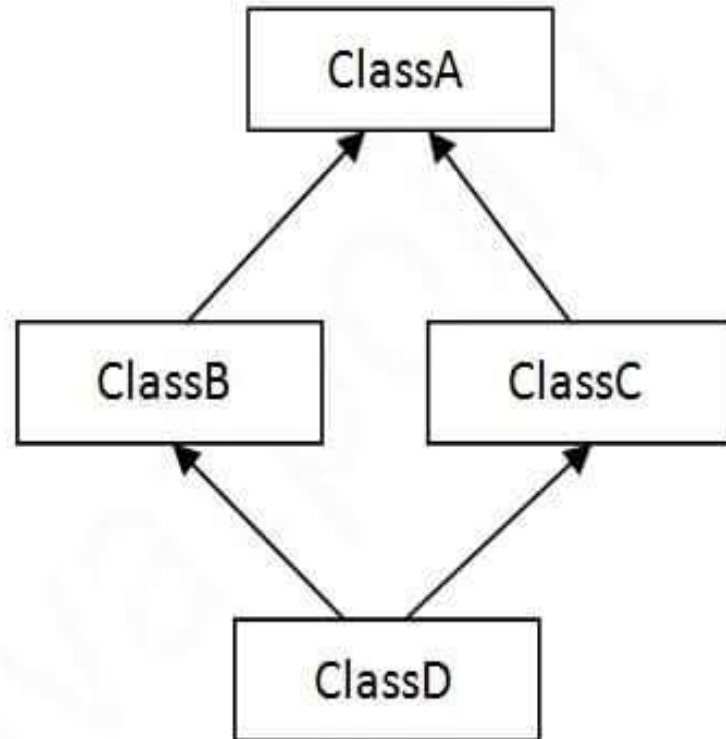
2) Multilevel



3) Hierarchical



4) Multiple



5) Hybrid

Polymorphism

- Poly – many
- Morphism – forms
- If one task is performed in different ways, it is known as polymorphism.
- **Polymorphism** is the ability of an object to take on many forms.
- Eg: Draw →