→ Acquiring the properties of one class in another class is called as inheritance.

→ Here property of the class indicates the members of the respective class. Members can be any thing like the data members or member functions.

→ Members can be accessible based on the access modifier. Private members can't be accessible outside of the defined class.

→ Here the class from which the members should be accessible is basically called as parent / base / super class

→ The class which is trying to access the members of the parent class is called as child / sub class

→ In case of interitance always the relationship between the parent & child is having "is-A" relationship.

Ex:

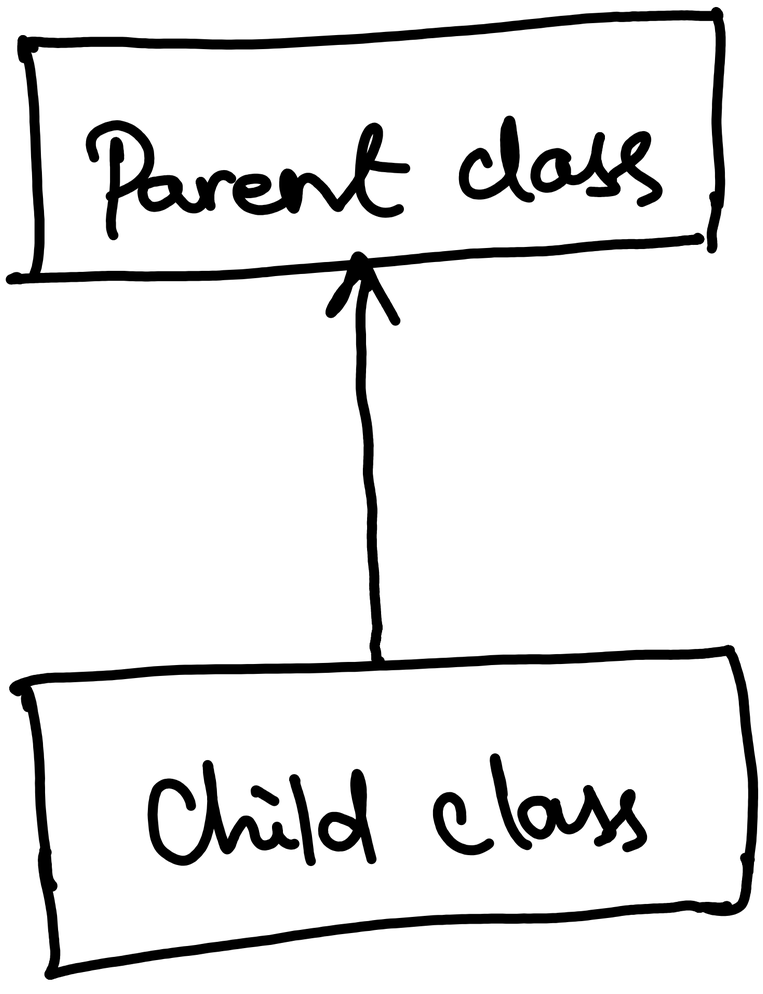
1. If we consider vehicle is the parent class & car is the child class then we say car is a vehicle

2. Similarly, iphone is a mobile

Child ↓ parent

relationship between parent child

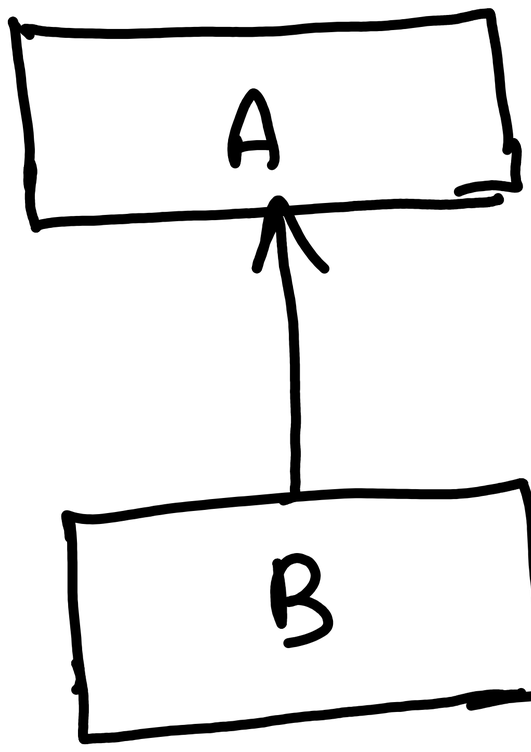
→ In Java class, the parent & child class are denoted diagrammatically as below



Always the arrow heading towards is treated as parent class & the arrow starts from the child class

Types of inheritance:

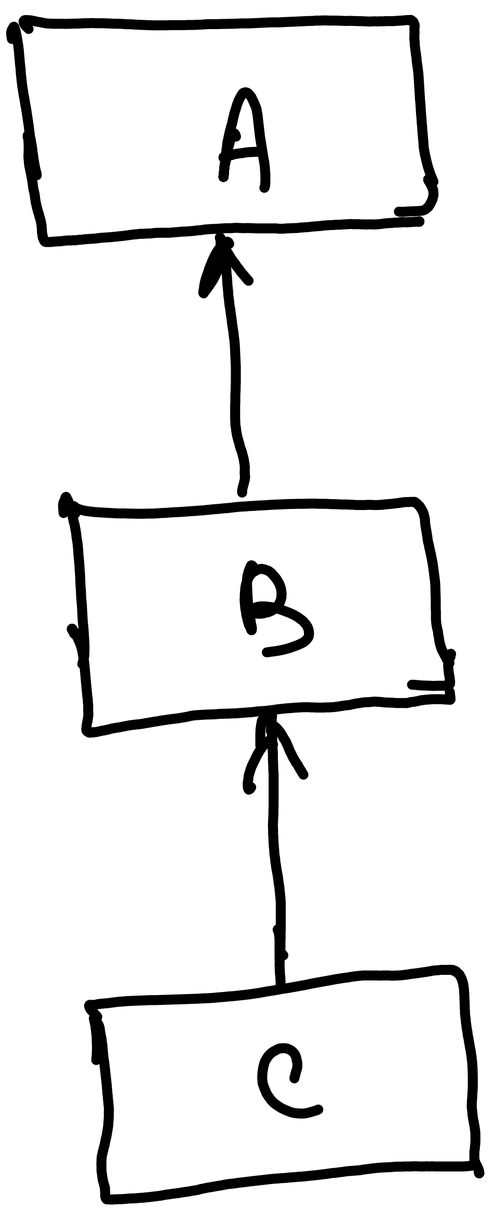
1.



Here Class 'B' can accels the members of class 'A'

(Single Level Inheritance)

2.

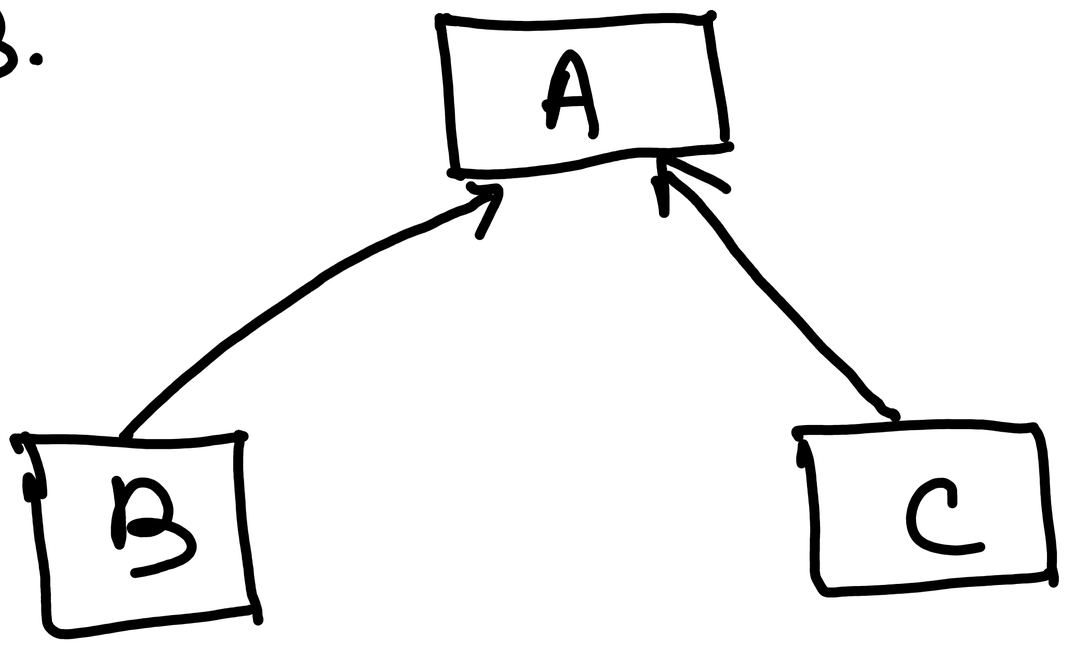


→ Here class 'B' can access the members of class 'A'

→ class 'C' can access the members of class 'B' & can also access the members of class 'A' indirectly via class 'B'

(Multi-level Inheritance)

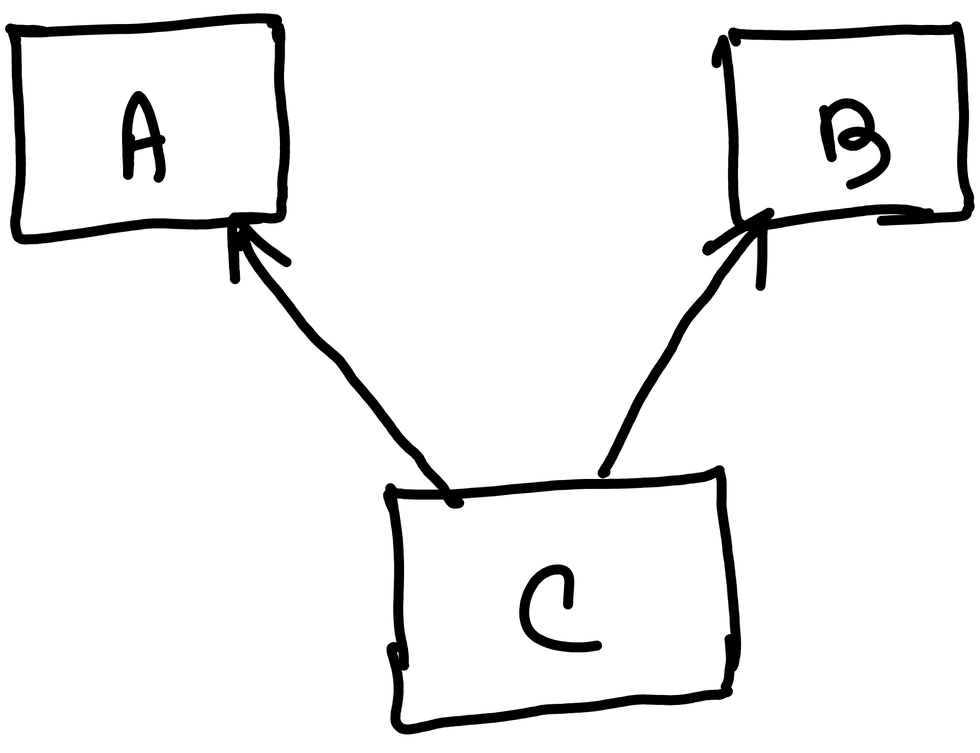
3.



(Hierarchial Inheritance)

Here both class 'B' & class 'C' can access the members of class 'A'

4.



(Multiple Inheritance)

Note:

In Java, multiple inheritance is supported in classes. If we still want to achieve the same then we've to use the interface concept