In overriding method resolution always talks care by jvm based on runtime objects add hence overriding is also considered as runtime polymorphism or Dynamic polymorphism

Rules for overriding

\* In overriding method name and arguments types must be match that is method signature must be same

\* In overriding return type must be same but this rule is applicable until 1.4 version only from 1.5 version we can take co-variant return type according to this child class return type need not be same as parent method return type it child type also allowed

Parent class method return type :-

Object ----------------> object /string/stringbuffer //true

Number ----------------> Num/Integer // true

String ----------------> object // false

Double ----------------> int // false

Co-variant return type concept applicable only for object type but not for primitive types

* Parent class private method not available to the child and hence overriding concept not applicable for private method
* Based on over requirement exactly same private method in child class it is valid but not overriding
* We can’t override final method in child classes
* Parent class abstract method we should override in child class to provide implementation
* We can override non-abstract method as abstract - the main advantage of we can stop the availability of parent method implementation to the next level child classes

Final <----------------> non-final // false

Non-final <----------------> final //true

Abstract <----------------> non-abstract //true

Synchronized <----------------> non- Synchronized //true

Native <----------------> non-native //true

Strictfp <----------------> non-strictfp true

While overriding we can’t reduce scope accesses modifier but we can increase scope modifier

Private < default < protected < public //increasing scope

Public - public

Protected - protected/public

Default - default/ protected/public

Inheritance

Is-A relationship

multiple inheritance:- java can't provide multiple inheritance because java does not extends more then at a time in classes

class A extends B,C // error

If over class don’t extends other then over class direct child class of object

Class A // A is child of object

* Directly and indirectly java won’t provide support for inheritance with respect to classes

Why java Won’t provide multiple inheritance

* Because Ambiguity problem java not provide
* But interface can extends any number of interface simultaneously hence java support multiple inheritance with respect to interface

Cyclic inheritance is not allowed in java because this is not required

Has-A relationship :-

Has-a relationship is also known as composition and Aggregation