Method Overriding

- "Writing two or more methods in super and sub classes with the same name and same signature is called Method Overriding".
- It is also known as '**late-binding' or 'run-time'** polymorphism.
- The method present in super class is called overridden method and the method present in the sub class is called over ridding method.
- When an overridden method is called through a super class reference, Java determines which version of that method is to be executed based upon the type of the object being referred to at the time the call occurs. Thus, this determination is made at run time. That is the reason it is called Run-time polymorphism.

Overridden Rules

- The argument list must exactly match that of the overridden method.
- The return type should exactly match that of the overridden method (upto 1.4V).
- After jdk 1.5 return types may not be same in co-variant return types. (Co-variant return type concept is applicable **only for object types** but **not for primitives)**
- The access level must not be more restrictive that that of the over ridden method. (private < default < protected < public)
- If a method can't be inherited we cannot override it (Ex: private).

- We can't override a static method as non static methods and vice versa.
- If you are over ridding two static methods then it will be "method hiding".
- The overriding method must not throw new broader checked exceptions than those declared by the overridden method .
- For unchecked exceptions there are no restrictions.

Overloading Vs Overriding

Method Overloading	Method Overriding
It occurs with in the same class	It occurs with in the super class and sub class
Inheritance is not involved since it deals with only one class	Inheritance is involved because it occurs between Super and Sub classes
In Overloading Return type need not be the same	In Overriding Return type must be same
Parameters must be different when we do Overloading	Parameters must be same
In Overloading one method can't hide another method	In Overriding sub class method hides the super class methods