Rules for method overloading:

- Overloaded methods MUST change the argument list
- Overloaded methods CAN change the return type
- Overloaded methods CAN change the access modifier
- Overloaded methods CAN declare new or broader checked exceptions
- A method can be overloaded in the same class or in a subclass

Rules for method overriding:

- In java, a method can only be written in Subclass, not in same class.
- The argument list should be exactly the same as that of the overridden method.
- The return type should be the same or a subtype of the return type declared in the original overridden method in the super class.
- The access level cannot be more restrictive than the overridden method's access level.
    - For example: if the super class method is declared public then the over-ridding method in the sub class cannot be either private or protected.
- Instance methods can be overridden only if they are inherited by the subclass.
- A method declared final cannot be overridden.
- A method declared static cannot be overridden but can be re-declared.
- If a method cannot be inherited then it cannot be overridden.
- A subclass within the same package as the instance's superclass can override any superclass method that is not declared private or final.
- A subclass in a different package can only override the non-final methods declared public or protected.
- An overriding method can throw any uncheck exceptions, regardless of whether the overridden method throws exceptions or not.
    - However the overriding method should not throw checked exceptions that are new or broader than the ones declared by the overridden

method. The overriding method can throw narrower or fewer exceptions than the overridden method.

- Constructors cannot be overridden.

Reference - https://crunchify.com/