**Method Overloading in Java**

Method Overloading is a feature that allows a class to have more than one method having the same name, if their argument lists are different.

**Ways to overload a method:**

1.Number of parameters.

add(int, int)

add(int, int, int)

2. Data type of parameters.

add(int, int)

add(int, float)

3. Sequence of Data type of parameters.

add(int, float)

add(float, int)

**Invalid case of method overloading:**

1.same name but differnent return type.

int add(int, int)

float add(int, int)

**Example:**

1. **public** **class** Listbox {
3. Select list;
5. **public** Listbox(Select list) {
6. **this**.list = list;
7. }
9. **public** **void** **select**(**int** i) {
10. **this**.list.selectByIndex(i);
11. }
13. **public** **void** **select**(String text) {
14. **this**.list.selectByVisibleText(text);
15. }


19. //other methods
21. }

**Method overriding**

Declaring a method in **sub class** which is already present in **parent class** is known as method overriding.

Overriding is done so that a child class can give its own implementation to a method which is already provided by the parent class.

Parent class method - Overridden method

Child class method - Overriding method

It is an example of **runtime polymorphism.**

To access the method in parent class we use the keyword super().

**Example:**

class Parent{

//Overridden method

public void disp()

{

System.out.println("disp() method of parent class");

}

}

class Child extends ABC{

//Overriding method

public void disp(){

System.out.println("disp() method of Child class");

}

super.disp();

public void newMethod(){

System.out.println("new method of child class");

}

public static void main( String args[]) {

/\* When Parent class reference refers to the parent class object

\* then in this case overridden method (the method of parent class)

\* is called.

\*/

Parent obj = new Parent ();

obj.disp();

/\* When parent class reference refers to the child class object

\* then the overriding method (method of child class) is called.

\* This is called dynamic method dispatch and runtime polymorphism

\*/

Parent obj2 = new Child();

obj2.disp();

}

}

**Rules of method overriding in Java**

1. Argument list

2. Access modifiers of child must not be more restrictive than parent.

3. private, static and final methods cannot be overridden as they are local to the class.

Example:

**Encapsulation in Java**

The whole idea behind encapsulation is to hide the implementation details from users. Also called as Data hiding.

How to implement encapsulation in java:  
1) Make the instance variables private.

2) Have getter and setter methods in the class to set and get the values of the fields.

Examples:

Reading from excel, xml.