**Method Overloading**

Process of defining methods in a class with the same name but different arguments called method overloading. Also called method overloading with argument.

Or

Name same, argument same but data type is different in parameter. This is called method overloading with data type.

Or

Multiple constructor of a class with same name but different argument, parameter or its datatype called constructor overloading.

**Example:**

class Emp{

// Constructor Overloading without Argument

Emp(){

System.out.println("Constructor without Argument");

}

Emp(String data){

System.out.println(data);

}

// Method Overloading with Argument (Single Argument)

void getName(){

System.out.println("Get name method without argument");

}

void getName(String name){

System.out.println(name);

}

void getName(String fName, String lName){

System.out.println(fName + " " + lName);

}

// Method overloadign with DataType (String)

void getAge(int age){

System.out.println("int age: " + age);

}

void getAge(String age){

System.out.println("String age: " + age);

}

}

public class Main {

// main Method

public static void main(String[] args) {

Emp emp = new Emp();

emp.getName("Chandan");

emp.getAge("23");

}

}

**Output:**Constructor without Argument

Chandan

String age: 23

**Method Overloading with Inheritance**

**Example:**

class User {

void getLogin(String name) {

System.out.println(name + "User Loged-in");

}

}

class Emp extends User {

***// Method overloading with inheritance on the basis of datatype. Although there is same parameter with different datatype.***

void getLogin(int password) {

System.out.println("Employee Loged-in");

}

}

public class Main {

public static void main(String[] args) {

Emp emp = new Emp();

emp.getLogin(2023);

}

}

**Output:**Employee Loged-in