**Method Overloading**

**1. Write two methods with the same name but different number of parameters of same type and call the methods from main method**

import java.io.\*;

class Addition {

public int add(int a, int b)

{

int sum = a + b;

return sum;

}

public int add(int a, int b, int c)

{

int sum = a + b + c;

return sum;

}

}

class Glf {

public static void main(String[] args)

{

Addition ob = new Addition();

int sum1 = ob.add(1, 2);

System.out.println("sum of the two integer value :"

+ sum1);

int sum2 = ob.add(1, 2, 3);

System.out.println(

"sum of the three integer value :" + sum2);

}

}

**3. Write two methods with the same name and same number of parameters of same type and call from main method**

class MethodOverloading {

private static void display(int a){

System.out.println("Arguments: " + a);

}

private static void display(int a, int b){

System.out.println("Arguments: " + a + " and " + b);

}

public static void main(String[] args) {

display(1);

display(1, 4);

}

}

**4. Write two methods with the same name and same number of parameters of different type and call from main method**

class MethodOverloading {

// this method accepts int

private static void display(int a){

System.out.println("Got Integer data.");

}

// this method accepts String object

private static void display(String a){

System.out.println("Got String object.");

}

public static void main(String[] args) {

display(1);

display("Hello");

}

}

**5. Write two methods with the same name, number of parameters and data type but different return Type.**

class HelperService {

private String formatNumber(int value) {

return String.format("%d", value);

}

private String formatNumber(double value) {

return String.format("%.3f", value);

}

private String formatNumber(String value) {

return String.format("%.2f", Double.parseDouble(value));

}

public static void main(String[] args) {

HelperService hs = new HelperService();

System.out.println(hs.formatNumber(500));

System.out.println(hs.formatNumber(89.9934));

System.out.println(hs.formatNumber("550"));

}

}