**Method Overloading in Java**

If a [class](https://www.javatpoint.com/object-and-class-in-java) has multiple methods having same name but different in parameters, it is known as **Method Overloading**.

## How to perform method overloading in Java?

Here are different ways to perform method overloading:

### Overloading by changing the number of parameters

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);

}

}

Output:

Arguments: 1

Arguments: 1 and 4

### Method Overloading by changing the data type of parameters

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");

}

}

**Output**:

Got Integer data.

Got String object.

Here, both overloaded methods accept one argument. However, one accepts the argument of type int whereas other accepts String object.

**Let’s look at a real-world example:**

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"));

}

}

**When you run the program, the output will be:**

500

89.993

550.00

### Important Points

* Two or more methods can have the same name inside the same class if they accept different arguments. This feature is known as method overloading.
* Method overloading is achieved by either:
  + changing the number of arguments.
  + or changing the data type of arguments.
* It is not method overloading if we only change the return type of methods. There must be differences in the number of parameter

