Method Overloading in Java

If a class has multiple methods having same name but different in parameters, it is known as Method Overloading.

If we have to perform only one operation, having same name of the methods increases the readability of the program
.
Suppose you have to perform addition of the given numbers but there can be any number of arguments, if you write the method such as a(int,int) for two parameters, and b(int,int,int) for three parameters then it may be difficult for you as well as other programmers to understand the behavior of the method because its name differs.
So, we perform method overloading to figure out the program quickly.
Advantage of method overloading
Method overloading increases the readability of the program.

Different ways to overload the method
There are two ways to overload the method in java

1. By changing number of arguments
2. By changing the data type

1) Method Overloading: changing no. of arguments

In this example, we have created two methods, first add() method performs addition of two numbers and second add method performs addition of three numbers.

In this example, we are creating static methods so that we don't need to create instance for calling methods.

```
class Adder
{
        static int add(int a,int b)
        {
                return a+b;
        }
        static int add(int a,int b,int c)
        {
                return a+b+c;
        }
}
class Overloading1
{
        public static void main(String[] args)
        {
                System.out.println(Adder.add(11,11));
                System.out.println(Adder.add(11,11,11));
        }
}
```

2. Method Overloading: changing data type of arguments
In this example, we have created two methods that differs in data type. The first add method receives two integer arguments and second add method receives two double arguments.

```java
class Adder
{
        static int add(int a, int b)
        {
                return a+b;
        }
        static double add(double a, double b)
        {
                return a+b;
        }
}
class Overloading2
{
        public static void main(String[] args)
        {
                System.out.println(Adder.add(11,11));
                System.out.println(Adder.add(12.3,12.6));
        }
}
```