JAVA FUNDAMENTALS COURSE

# METHODS OVERLOADING

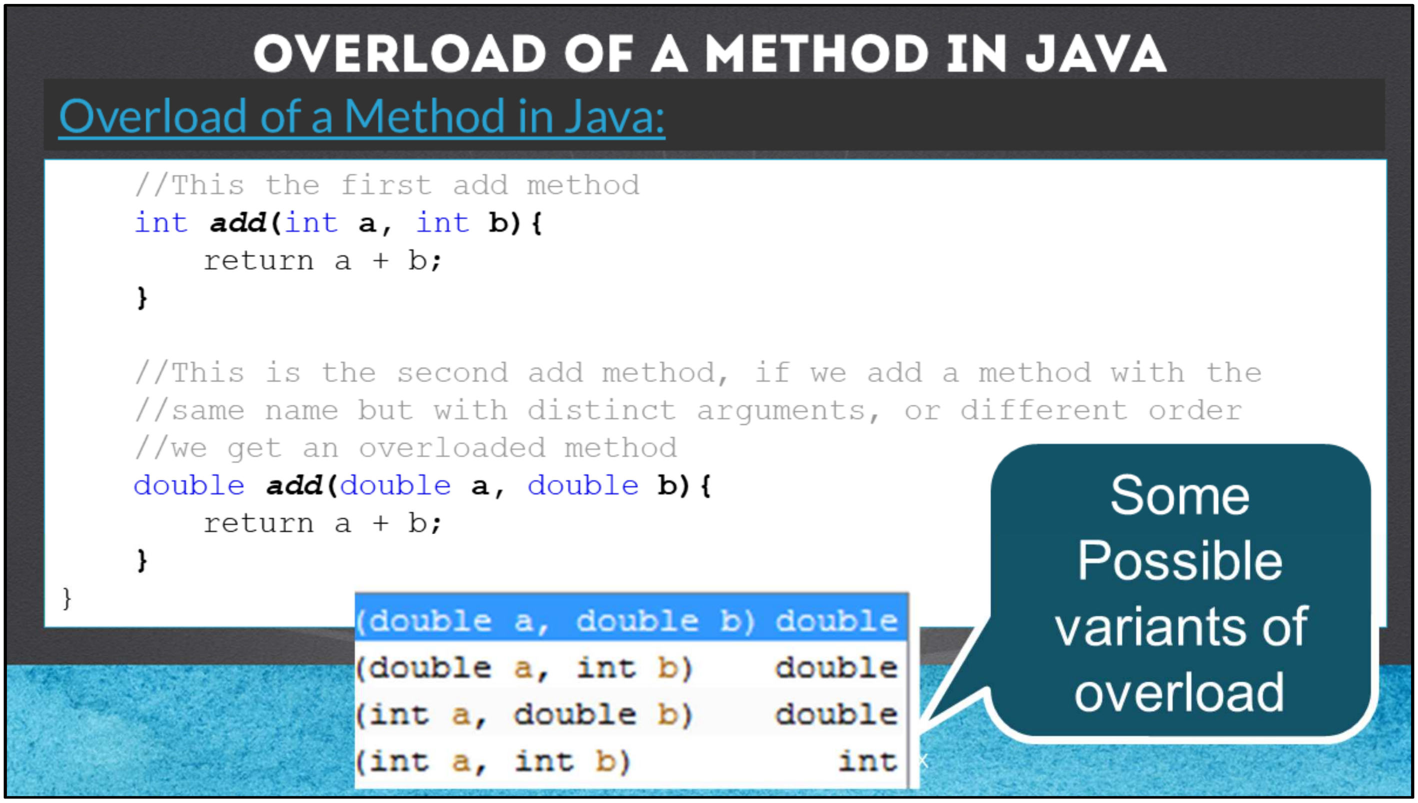
By the expert: Ubaldo Acosta

JAVA FUNDAMENTALS COURSE
www.globalmentoring.com.mx

Hello, Ubaldo Acosta greets you again. I hope you're ready to start with this lesson.

We are going to study the topic of overloading methods in Java.

Are you ready? Come on!



OVERLOAD OF A METHOD IN JAVA

Overload of a Method in Java:

```
    //This the first add method
    int add(int a, int b){
        return a + b;
    }

    //This is the second add method, if we add a method with the
    //same name but with distinct arguments, or different order
    //we get an overloaded method
    double add(double a, double b){
        return a + b;
    }
}
```

```
(double a, double b) double
(double a, int b)    double
(int a, double b)    double
(int a, int b)          int
```

Some Possible variants of overload

The overload of methods in Java is similar to the issue of Constructor overload, but in this case any method can be overloaded.

The overload of methods is to offer more than one option of any of the methods defined in our classes, in order to provide more options for the use of our methods.

In the code shown we can see the example of the add method. First we define a first add method, which receives two arguments of type int, until now there is no overload, because for that there must be two or more methods with the same name defined in the class.

One of the rules of overloading methods is that what the compiler observes for a method to perform a valid overload is that the types of the arguments are different from those of the method already defined, this includes the order of the arguments, but in NO case the compiler check that the names of the arguments are the same or not, likewise it does not check whether the return type is the same or not, that is, that the return type does not matter at the time of adding a method that meets with the overload.

So, for an overload to be valid, it must comply with the following:

1) The name of the method must be the same as the method that you want to overload.

2) The arguments of the method must be different to the method that you want to overload, only the type and order in which they are added are reviewed, the name of the argument is not reviewed.

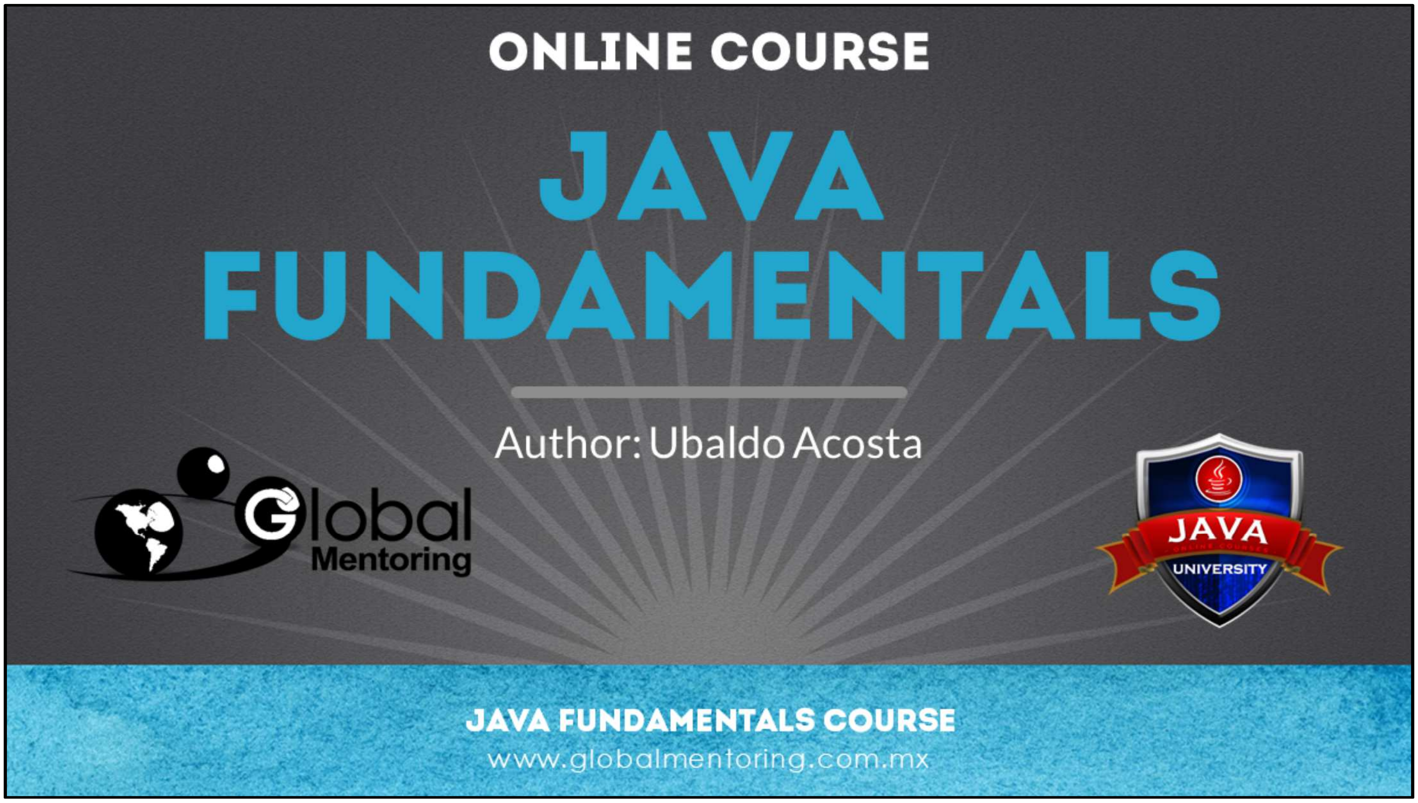
3) The type of return does not affect if it is the same or different from the method to overload.

4) The method to overload can be defined in our class or in some Parent class.

5) Arguments can be primitive or Object type

In case of applying any conversion of the types to be used, the higher automatic conversion applies, for example if we use two variables of type long, and we make a call to the add method, the method will not be called with arguments of type int, but to the method add with arguments of type double, since the type long will be converted to the superior type automatically, that is to say the double type, and the int type will be discarded because it is a type of smaller bits and less able to store a data of type long. The same applies to the concept of objects, although this topic will be seen in the next level, in the topic of object conversion.

Basically with this is enough to know what is the overload and how to implement it in Java, then we will perform an exercise.



ONLINE COURSE

# JAVA FUNDAMENTALS

Author: Ubaldo Acosta

JAVA FUNDAMENTALS COURSE

www.globalmentoring.com.mx

Global Mentoring