Yes, we can overload the method with same return type but with different number of arguments. Refer EX:1 where we have two add methods with same return type **int** but one with 2 and another with 3 arguments.

There are two ways to overload the method in java

1)By changing number of arguments

2)By changing the data type

**1) By changing number of arguments - but same return type int :**

Ex:1

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 TestOverloading1{

public static void main(String[] args){

System.out.println(Adder.add(11,11));

System.out.println(Adder.add(11,11,11));

}}

**Output:**

22

33

**2)By changing the data type**

Ex:2

class Adder{

static int add(int a, int b){return a+b;}

static double add(double a, double b){return a+b;}

}

class TestOverloading2{

public static void main(String[] args){

System.out.println(Adder.add(11,11));

System.out.println(Adder.add(12.3,12.6));

}}

**output:**

22

24.9

* **In java, method overloading is not possible by changing the return type of the method only because of ambiguity.**

Ex:3

class Adder{

static int add(int a,int b){return a+b;}

static double add(int a,int b){return a+b;}

}

class TestOverloading3{

public static void main(String[] args){

System.out.println(Adder.add(11,11));//ambiguity

}}

**Output:**

**Compile Time Error: method add(int,int) is already defined in class Adder**