**Method Overloading Example**

**Example part1:**

Create class Addition that with following methods:

* method with name add that takes an integer parameter and returns parameter value+10.
* method with name add that takes two integer parameters and returns the summation of these parameters.
* method with name add that takes three integer parameters and returns the summation of these parameters.
* method with name add that takes two parameters first one is integer and second one is double and returns the summation of these parameters+10.
* method with name add that takes two parameters first one is double and second one is integer and returns the summation of these parameters+20.
* method with name add that takes two double parameters and returns the summation of these parameters+30.

**sol:**

**public class Addition {**

**public int add(int num)**

**{**

**return num+10;**

**}**

**public int add(int num1,int num2)**

**{**

**return num1+num2;**

**}**

**public int add(int num1,int num2,int num3)**

**{**

**return num1+num2+num3;**

**}**

**public int add(int num1,double num2)**

**{**

**return num1+(int)num2+10;**

**}**

**public int add( double num1,int num2)**

**{**

**return (int)num1+num2+20;**

**}**

**public int add( double num1,double num2)**

**{**

**return (int)num1+(int)num2+30;  }**

**}**

**Example part2:**

In the main method, write code to create an object from class Addition  then use the instance method add using this object to find and print the summation for each of the following: (what is the output in each case?)

* 30.
* 10, 20, 30
* 10, 20
* 10, 20.0
* 10.0, 20
* 10.0, 20.0

**Solution**

**public static void main(String[] args) {**

 **Addition a1 = new Addition();**

 **System.out.println("add(30)= "+a1.add(30));**

 **System.out.println("add(10,20,30)= "+a1.add(10,20,30));**

 **System.out.println("add(10,20)= "+a1.add(10,20));**

 **System.out.println("add(10,20.0)= "+a1.add(10,20.0));**

 **System.out.println("add(10.0,20)= "+a1.add(10.0,20));**

 **System.out.println("add(10.0,20.0)= "+a1.add(10.0,20.0));**

 **}**

output



**Example part3:**

in class Addition  create a new method with name add that takes two integer parameters and returns the parameter summation as double.

Did you succeed? Why?

**No, since it has the same parameters list (type, number) as add method defined in Example part1.**

**Notes:**

* Method overloading happens when a class contains more than one method having the same name, but with a different parameter list.
* Any change in parameters number, type or order means that the parameter list is different.
* method name and parameter list called method signature.
* Returned data type of the method has nothing to do with method overloading
* Creating two methods in the same class with the same parameters number and type but with different return types will generate an error since return type is not part of method signature.
* The concept of overloading applies also to constructors as they are special kinds of methods.
* It is possible to have more than one constructor with different parameter list.

