S.No: 3	Exp. Name: ***Write a Java program to implement Method overloading***	Date:2025-09-16

Aim:

Write a Java program with a class name `Addition` with the methods `add(int, int)`, `add(int, float)`, `add(float, float)` and `add(float, double, double)` to add values of different argument types.

Write the **main(String[])** method within the class and assume that it will always receive a total of **6** command line arguments at least, such that the first **2** are **int**, next **2** are **float** and the last **2** are of type **double**.

If the **main()** is provided with arguments : **1, 2, 1.5f, 2.5f, 1.0, 2.0** then the program should print the output as:

```
Sum of 1 and 2 : 3
Sum of 1.5 and 2.5 : 4.0
Sum of 2 and 2.5 : 4.5
Sum of 1.5, 1.0 and 2.0 : 4.5
```

Note: Please don't change the package name.

Source Code:

q11266/Addition.java

```
package q11266;
import java.util.Scanner;
public class Addition{
   public void add(int a, int b){
      System.out.println("Sum of "+a+" and "+b+" : "+(a+b));
   }
   public void add(float a, float b){
      System.out.println("Sum of "+a+" and "+b+" : "+(a+b));
   }
   public void add(int a, float b){
      System.out.println("Sum of "+a+" and "+b+" : "+(a+b));
   }
   public void add(float a, double b, double c){
      System.out.println("Sum of "+ a + ", "+b+" and "+c+" : "+(a+b+c));
   }
   public static void main(String[] args){
      Addition obj = new Addition();
      int i1 = Integer.parseInt(args[0]);
      int i2 = Integer.parseInt(args[1]);

      float f1 = Float.parseFloat(args[2]);
      float f2 = Float.parseFloat(args[3]);

      double d1 = Double.parseDouble(args[4]);
      double d2 = Double.parseDouble(args[5]);
      obj.add(i1,i2);
      obj.add(f1,f2);
      obj.add(i2,f2);
      obj.add(f1,d1,d2);
```

```
    }
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Sum of 2 and 1 : 3
Sum of 5.0 and 3.6 : 8.6
Sum of 1 and 3.6 : 4.6
Sum of 5.0, 9.2 and 5.26 : 19.46