```java
//FLOAT AND DOUBLES

//When Java uses floating point numbers, it uses 2 types of values: float and double.
//The difference between the types is that, the float can store a smaller amount of numbers after the floating
//point, that means that the precision of the float is less than the precision of the double, because the
//double stores twice as many decimal places as the float can store.


public class FloatAndDoubles {

    public static void main(String[] args){

        //Heads up! Maximum float storage is seven, no matter if it is before or after floating
        //point, float always uses seven numbers and rounds off the last number.
        //Also, float always uses "f" at end of the numbers assigned to the variable...
        float numberDecimalPlaces = 77.52356559f; // => 77.52357 | the last number is round
        System.out.println(numberDecimalPlaces);

        //While, double can use twice as many numbers, and doesn't need to use "f", "d" or any character
        //representation. The maximum storage of double is 15 numbers, it doesn't matter if they are before
        //or after the floating point...
        double numberWithMoreDecimalPlaces = 77.1234567890123456; // => 77.12345678901235
        System.out.println(numberWithMoreDecimalPlaces);

        //Caution! When using integers with double operations, Java interprets numbers as integers
        //and gives unexpected results...
        double division = 5 / 2;
        System.out.println(division); // => 2.0 | When expected is 2.5...

        //Always use an operator like floating point to get the expected results...
        double divisionExpected = 5 / 2.0;
```

```
        System.out.println(divisionExpected); // => 2.5 | Expected result!

        //CAUTION!! When using double, some accounts may return unexpected results, for example...
        double sumWeird = 0.1 + 0.2; //We wait for result => 0.3, but...
        System.out.println(sumWeird); // => 0.30000000000000004 | What? This is because binary operations
                                      //in Java (and other languages) do not handle operations very well
                                      //with floating point numbers...
    }

}
```