

Question 7

**Java Supports Pass by value or
Pass by reference**



Master Java with Suraz Ghimire

- **12 years of experience in the Industry**
- **Mentoring Data Engineers since 2014**

**This is another Amazing
Interview Question.
Java supports pass by value or
pass by reference.**

Let's try to understand it today.



For Primitive Types:

Java Supports Pass by Value, which means whenever you pass the value to the method, a copy is created and its value would be passed.





```
package in.olc.utils;

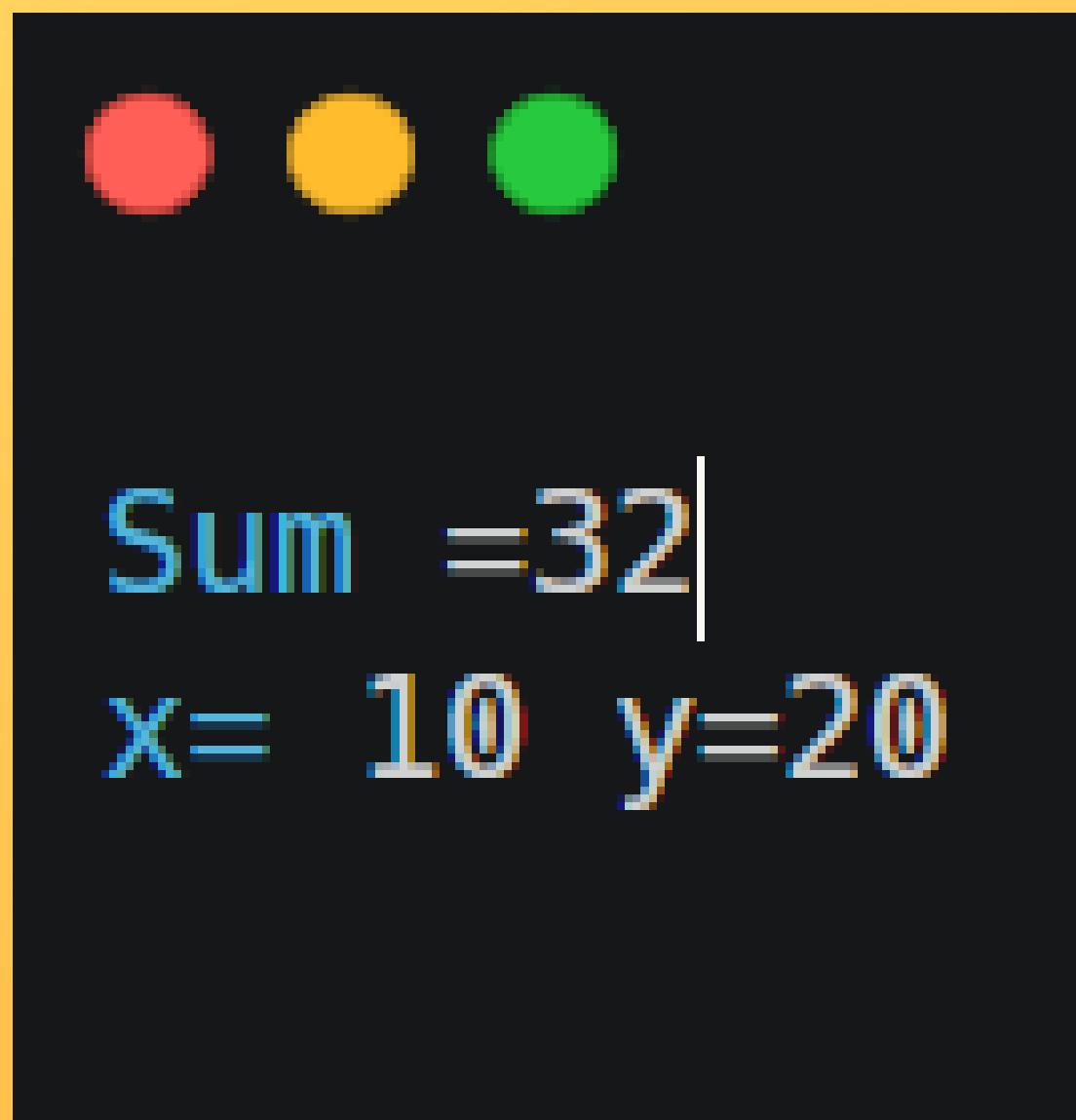
class PassByValue {
    public static void main(String[] args) {
        int x=10;
        int y=20;

        int output=add(x,y);
        System.out.println("Sum =" +output);
        System.out.println("x= "+x+" y=" +y);
    }

    public static int add(int num1,int num2){
        num1++;
        num2++;
        int result=num1+num2;
        return result;
    }
}
```



Output:



Here is the output of Java Visualizer.



The screenshot shows the Java Visualizer interface. At the top, there are tabs for Debugger, Console, and Java Visualizer (which is selected). Below the tabs, there are two main sections: "Call Stack" and "Objects".

Call Stack:

- main:8
 - x 10
 - y 20
- add:13
 - num1 10
 - num2 20

Objects:



Let's understand how it works for Object Types:

This is very tricky.

Lets understand with an
example





```
package in.olc.utils;

class Num{
    int a;
    int b;
}

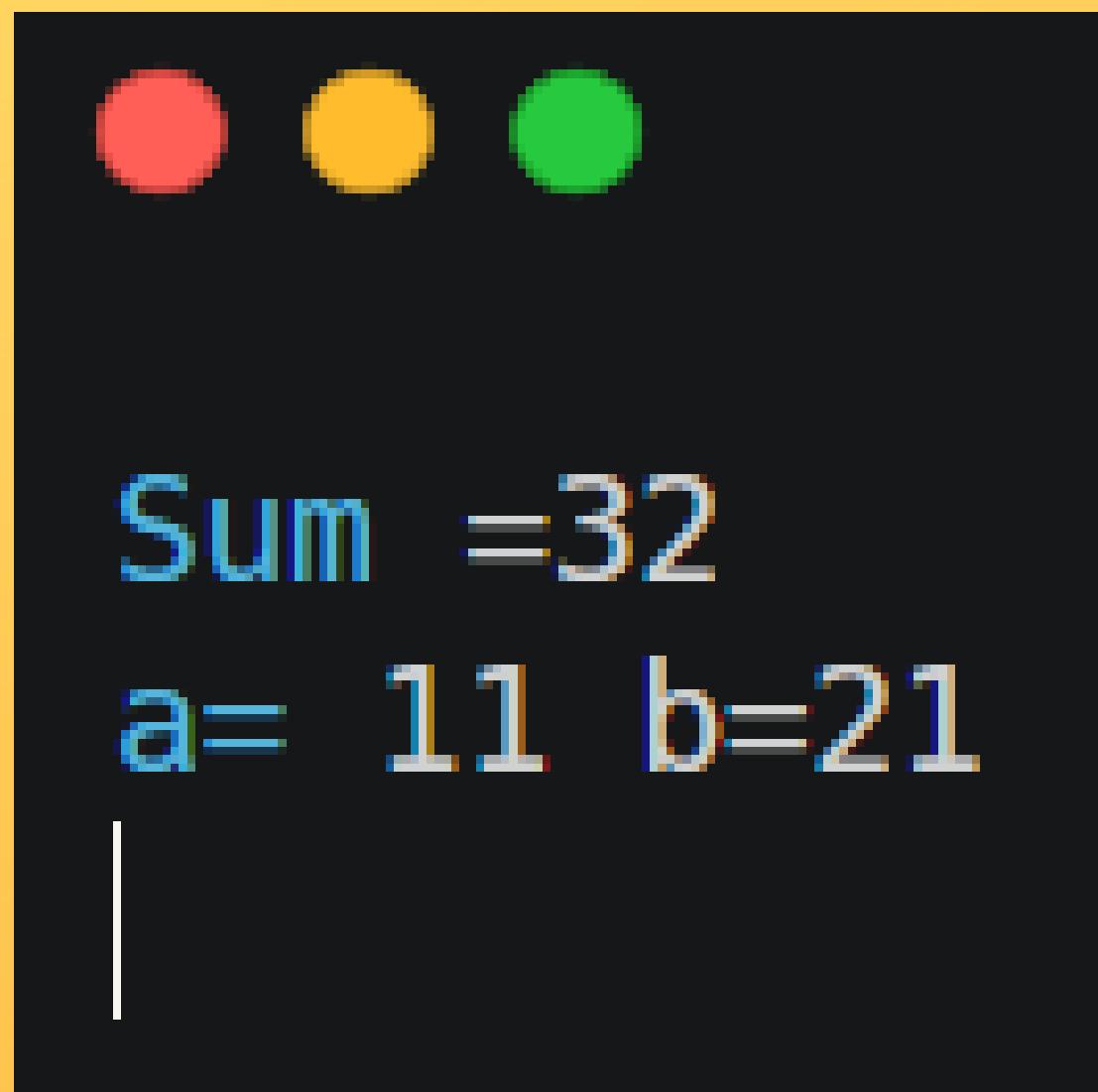
class PassByValue {
    public static void main(String[] args) {
        Num n1= new Num();
        n1.a=10;
        n1.b=20;

        int output=add(n1);
        System.out.println("Sum =" +output);
        System.out.println("a= "+n1.a+" b=" +n1.b);
    }
    public static int add(Num n2){

        n2.a++;
        n2.b++;
        int result=n2.a+n2.b;
        return result;
    }
}
```



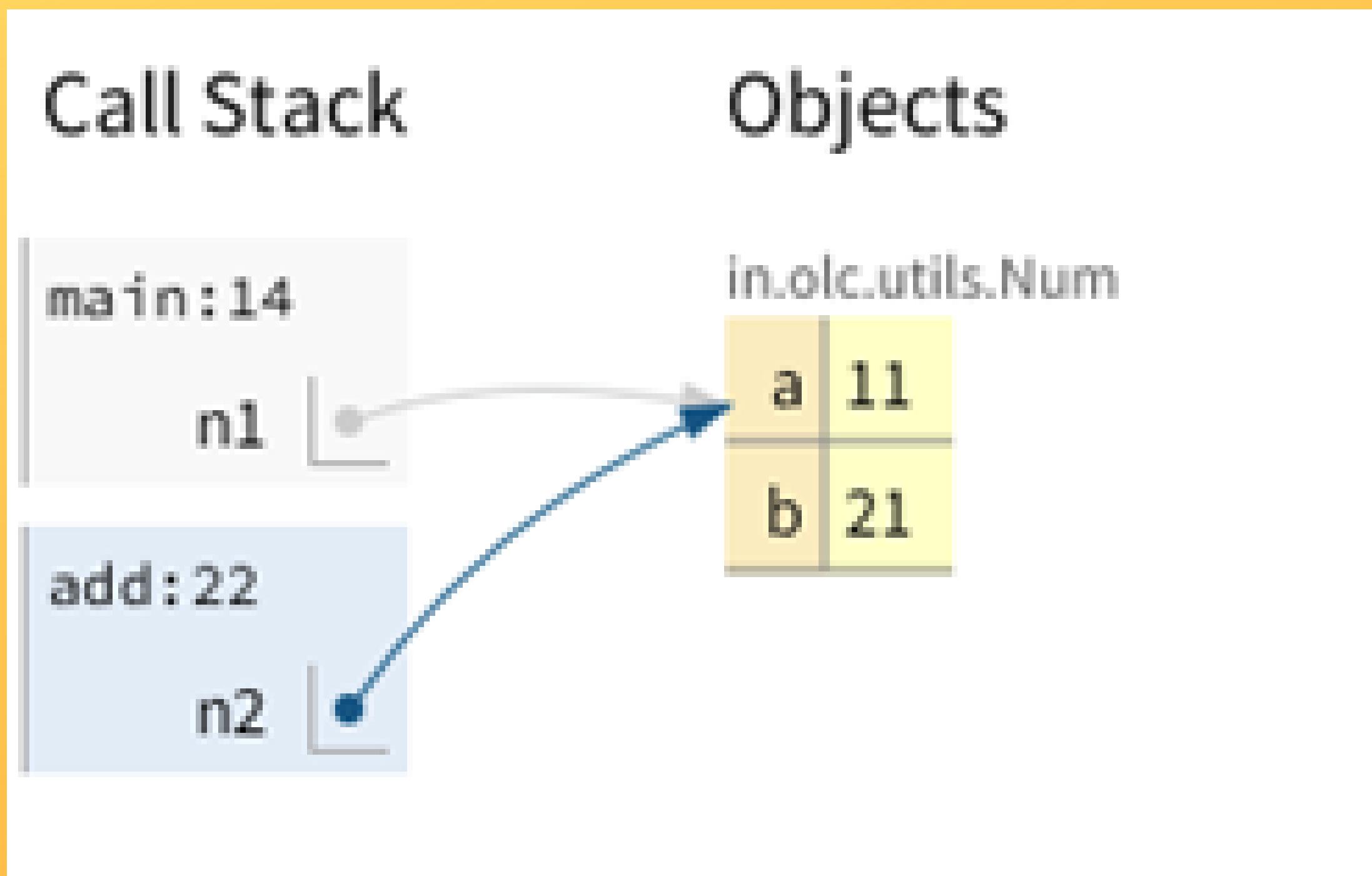
Output:



**Looks like,
it is a Pass by Reference**



The memory diagram shows something like this.



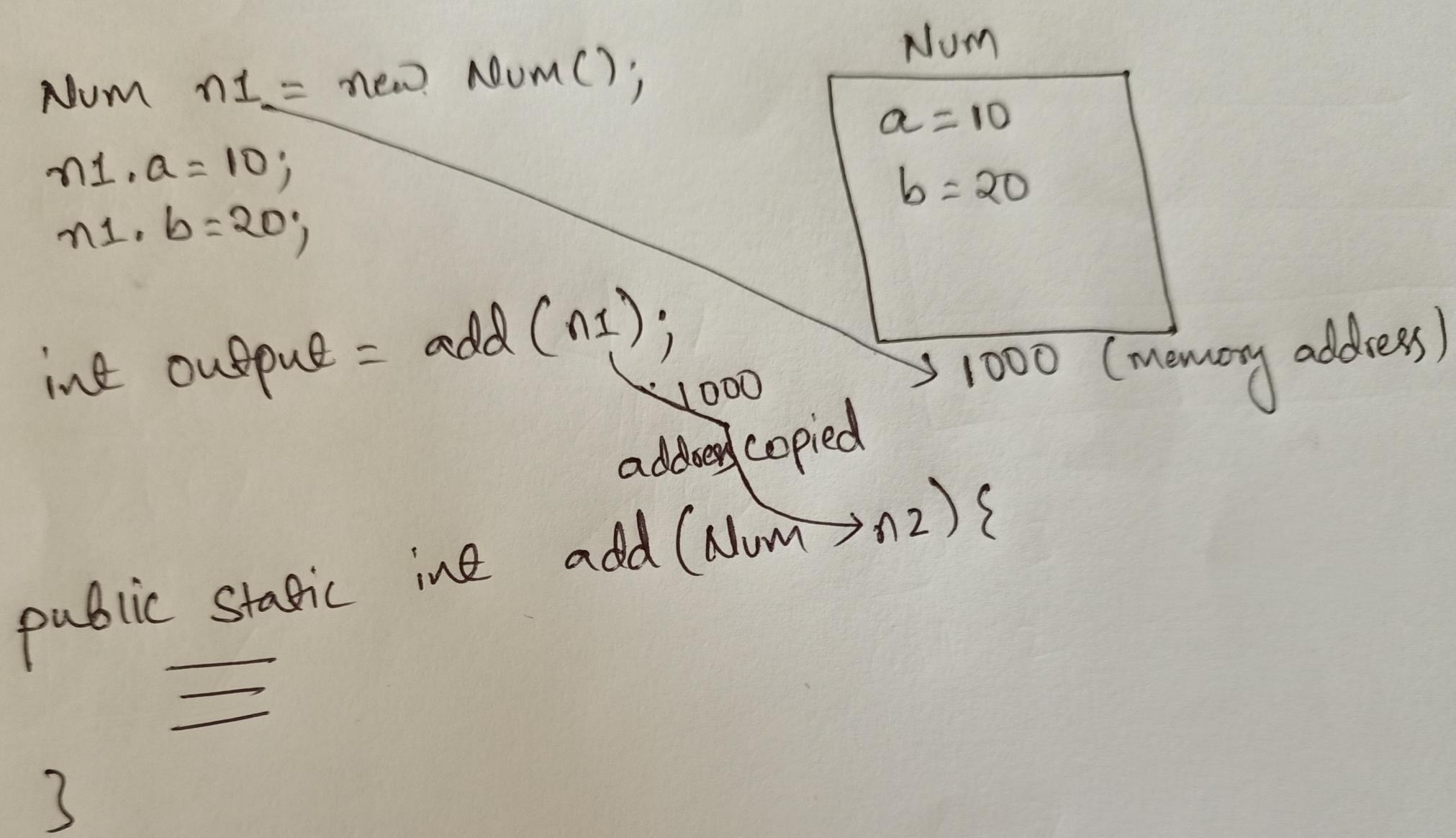
Fact:

The object of Num is created only once in the heap.

In the main method add(n1) will copy the value of n1(reference) and passes it to the called function.

Both of them would be ultimately pointing to the same Num Object. →

Assuming the memory address of Num Object is 1000



$n1=1000$,
 and this 1000 gets copied to $n2$,
 which points to same object



★ You can say it is a pass-by-reference (Just for your easiness) though it is actually a passed-by value where only the reference is copied and not the object.

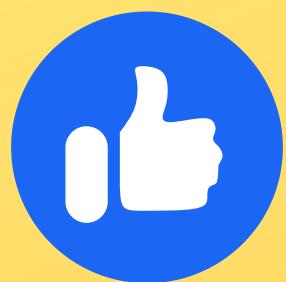
★ Both n1 and n2 point to the same object and hence the changes are done in the add() and reflect back to the object in the main method.



Conclusion:

**Java support pass by value only
(However, calling pass-by-
reference in case of an object will
not be so wrong)**





Stay tuned for more



<https://www.linkedin.com/in/surazghimire/>

Follow for more such content

If anyone says Java is tough, send them to OLC, I will make them a champion and love Java..



<https://www.youtube.com/@OnlineLearningCenterIndia>

**Master Java Programming at
www.onlinelearningcenter.in/**

Suraz Ghimire: Online Learning Center