Java is pass-by-value or pass-by-reference

<https://stacktraceguru.com/java/java-is-pass-by-value>

Most of the java developers get confused when it comes to the question that, is Java 'pass-by-value' or 'pass-by-reference'.

The answer is that**Java is pass-by-value**. We will verify that in this blog under the section [Pass-by-value for Object data types](https://stacktraceguru.com/java/java-is-pass-by-value#object-pass-by-value). Before jumping straight to the conclusion, let's understand in detail how different variables work in java.

Common mistake done by Java developers

There is one most common mistake that java developers do. They try to update object value using "=" or "new" keyword.

Let's look at an example where we want to change the color of box and square a number.

// Wrong - this will not return the updated value to calling method

public void changeToPink(Box boxToChange){

boxToChange = **new** Box("Pink");

}  
  
public void square(int value){

value **=**value\*value;

}

Topics we will be discussing

* [What is pass-by-value](https://stacktraceguru.com/java/java-is-pass-by-value#What-is-pass-by-value)
* [What is pass-by-reference](https://stacktraceguru.com/java/java-is-pass-by-value#What-is-pass-by-reference)
* [Pass-by-value for primitive data types](https://stacktraceguru.com/java/java-is-pass-by-value#primitive-pass-by-value)
* [Pass-by-value for Object data types](https://stacktraceguru.com/java/java-is-pass-by-value#object-pass-by-value)
* [If java is pass-by-value how objects are updated](https://stacktraceguru.com/java/java-is-pass-by-value#how-objects-are-updated)

What is pass-by-value?

Passing variable by value means the value of the variable is copied into another variable. This copies the data into new memory.

Hence modifying new variable only updates the copy. The original variable value is not changed.

We will look in detail with examples later in this blog.

What is pass by reference?

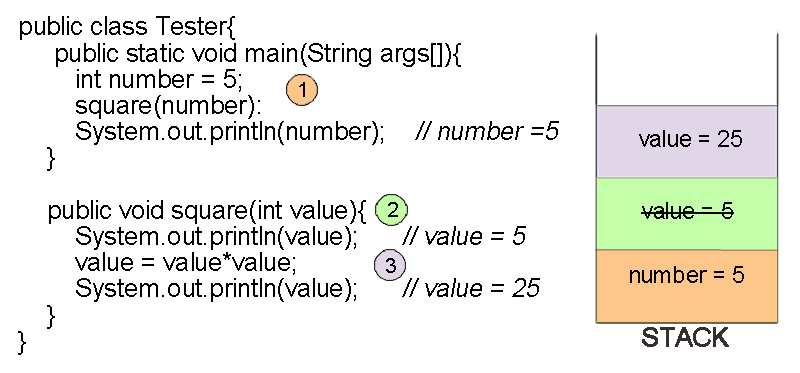
Passing variable by reference means actual object reference or pointer is passed.

This means modifying the new reference object will update the original object.

Pass-by-value in java for primitive data types

Primitive variables are not objects. So primitive data types are pass-by-value. Which means only value is passed to new variable.

Lets look at the example for better understanding



In the above code we can see that the value of number variable is 5. Then we call the square function.

Even after calling the square function value of number variable is 5 and does not change. Because, we just updated the local(new) variable in square function. This local variable is created in the scope of the function and hence the original number variable is not updated.

How to update primitive parameter from function call?

The only way to update the original primitive variable is by returning value from the function.

Following code will update the value of number variable after calling the function.

**public** **class** **Tester**{

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

int number = 5; // number =5

number = **square**(number):

**System**.out.**println**(number); // number = 25

}

**public** int **square**(int value){

value = value\*value; // value= 25

**return** value;

}

}

How objects are updated?

If we update the properties of object using any reference variable, values of all reference variables are updated.  
In other words if you **update the object not the reference variable** then original object is updated.

Example if we use following code to update the color of object userBox

**public** **class** **Box**{

**String** color;

**Box**(**String** color){

this.color = color;

}

}

**public** **class** **Tester**{

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

**Box** userBox = **new** **Box**("Purple");

**changeToPink**(userBox);

**System**.out.**println**(userBox); // Pink

}

**public** **static** void **changeToPink**(Box boxToChange){

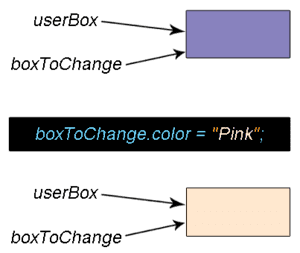
**System**.out.**println**(boxToChange); // Purple

boxToChange.color="Pink";

**System**.out.**println**(boxToChange); // Pink

}

}



Using above method the object userBox in main method is also updated to pink color.

Pass-by-value in java for Object data types

In java even objects are pass by value. In fact it is **passing reference as value**. Even if it sounds confusing don't worry, we can explain it.

In java every object has reference variables. So when we pass object to another function the reference variable is passed. So the references to the actual object is copied.  
Now two different reference variables point to the same object. Point to be noted here is that we have two different reference variables.

**public** **class** **Box**{

**String** color;

**Box**(**String** color){

this.color = color;

}

}

**public** **class** **Tester**{

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

**Box** userBox = **new** **Box**("Purple");

**changeToPink**(userBox);

**System**.out.**println**(userBox); // Purple

}

**public** void **changeToPink**(Box boxToChange){

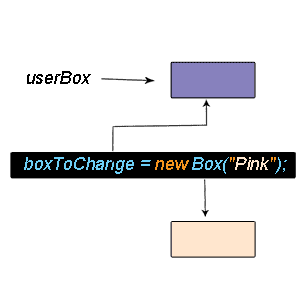
**System**.out.**println**(boxToChange); // Purple

boxToChange = **new** **Box**("Pink");

**System**.out.**println**(boxToChange); // Pink

}

}



In the above code we have object userBox with color "purple". Now call the function changeToPink(). Even after calling the function changeToPink(), color of the user box remains "Purple".

When you use new keyword new object is created. The new box with pink color is stored in local/new variable(boxToChange). Original object userbox remains with old color purple.

Conclusion

As we can see that the updated reference variable does not update the original object. So we can say when we call the function new reference to same objecect is created, so only reference address is passed as value .

Fast track reading

* Java is pass by value
* [Pass by value means value of the variable is copied into another variable](https://stacktraceguru.com/java/java-is-pass-by-value#What-is-pass-by-value)
* [Pass by reference means actual object reference or pointer is passed](https://stacktraceguru.com/java/java-is-pass-by-value#What-is-pass-by-reference)
* [Only way to update original primitive variable is by returning value from the function](https://stacktraceguru.com/java/java-is-pass-by-value#update-primitive-value)
* [In java passing reference copies the object reference as a value](https://stacktraceguru.com/java/java-is-pass-by-value#object-pass-by-value)