

Static Methods

optional for void

optional since this class contains only static methods

```
class MethodTest1 {
    /**/
    public MethodTest1 () { }
    /**/
    public static void method1 ( int i, int j ) {
        i = 5; j = 6;
        return;
    }
    /**/
    public static int method2 ( int i ) {
        i = 5;
        return i;
    }
}
```

parameter list

parameter

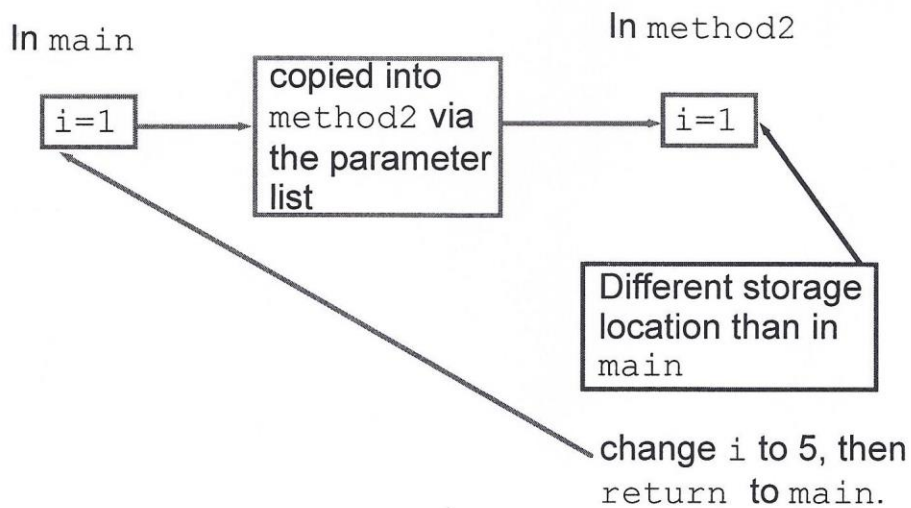
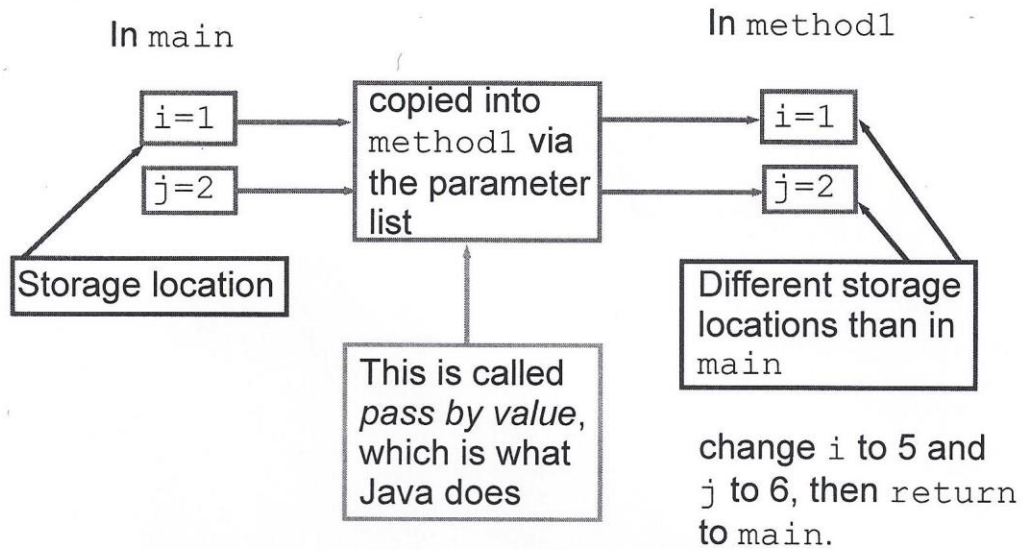
return type

mandatory

Class Work: Download MethodTest1.java from canvas.instructure.com.
Type in and run Test1.java. Talk about the results.

```
public class Test1 {
    /**/
    public static void main ( String [] arg ) {
        /**/
        int i = 1; int j = 2;
        System.out.println();
        System.out.println( "i = " + Integer.toString(i) );
        System.out.println( "j = " + Integer.toString(j) );
        MethodTest1.method1(i,j);
        System.out.println( "i = " + Integer.toString(i) );
        System.out.println( "j = " + Integer.toString(j) );
        /**/
        i = 1;
        System.out.println();
        System.out.println( "i = " + Integer.toString(i) );
        i=MethodTest1.method2(i);
        System.out.println( "i = " + Integer.toString(i) );
        /**/
        return;
    }
    /**/
    public Test1 () { }
}
```

static methods are invoked through their class name



```

class MethodTest2 {
    /**/
    public MethodTest2 () { }
    /**/
    public static void method3 ( int [] i ) {
        i[0] = 5; i[1] = 6;
        return;
    }
    /**/
    public static int [] method4 () {
        int [] j = new int [2];
        j[0] = 3; j[1] = 4;
        return j;
    }
}

```

preferable to method3.
It is more in the spirit
of the Java programming
language.

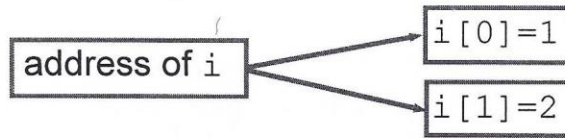
Class Work: Download MethodTest2.java from canvas.instructure.com.
Type in and run Test2.java. Talk about the results.

```

public class Test2 {
    /**/
    public static void main ( String [] arg ) {
        /**/
        int [] i = new int [2];
        i[0] = 1; i[1] = 2;
        System.out.println();
        System.out.println( "i[0] = " + Integer.toString( i[0] ) );
        System.out.println( "i[1] = " + Integer.toString( i[1] ) );
        MethodTest2.method3(i);
        System.out.println( "i[0] = " + Integer.toString( i[0] ) );
        System.out.println( "i[1] = " + Integer.toString( i[1] ) );
        /**/
        System.out.println();
        int [] j = MethodTest2.method4();
        System.out.println( "j[0] = " + Integer.toString( j[0] ) );
        System.out.println( "j[1] = " + Integer.toString( j[1] ) );
        /**/
        System.out.println();
        System.out.println(j);
        /**/
        return;
    }
    /**/
    public Test2 () { }
}

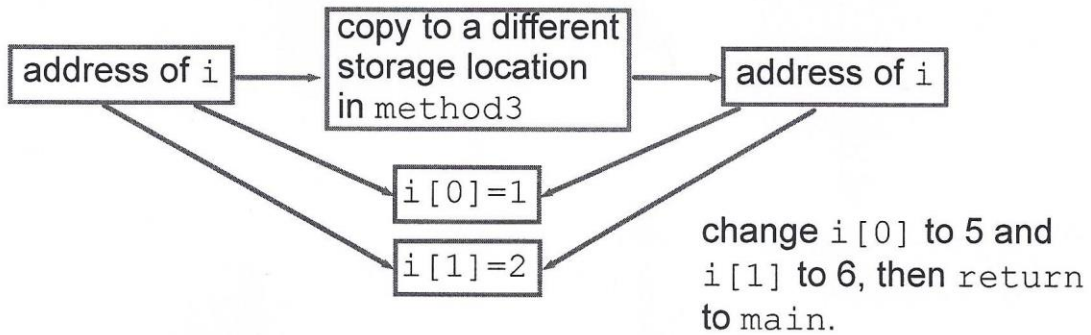
```

● Structure of an array



In main

In method3



In main

In method4

