CSSE220 - How to draw a boxes and pointers diagrams

1. Look at the provided code. For every variable in the code, draw a little box next to it.

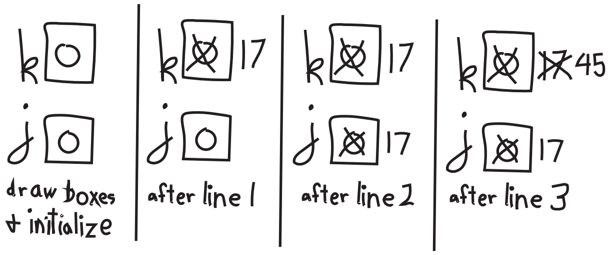
- Variable is a primitive type - then the box will contain the variable's actual value
- Variable is a non-primitive type, i.e., object - then the box will contain the tail of an arrow

Then hand trace the code line by line. What you do next depends on each line of code that you hand trace.

2. An assignment to a variable that holds a primitive type

If the code contains an assignment to a variable that holds a primitive type, then a copy of the value is placed in the variable's box. So, the variable being assigned to (on the left-hand side of the assignment statement) receives a copy of the value from the right-hand side of the assignment statement.

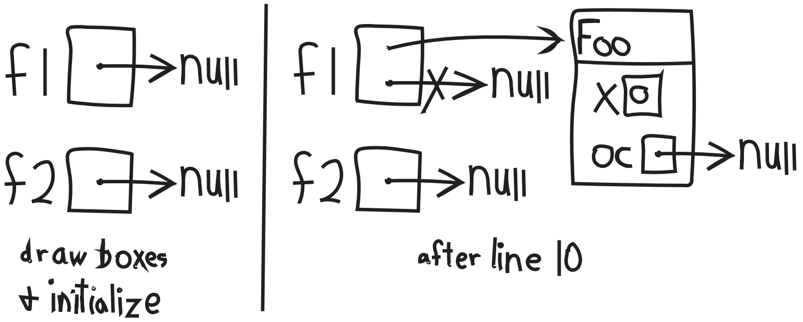
```
1   int k = 17;
2   int j = k;
3   k = 45;
```

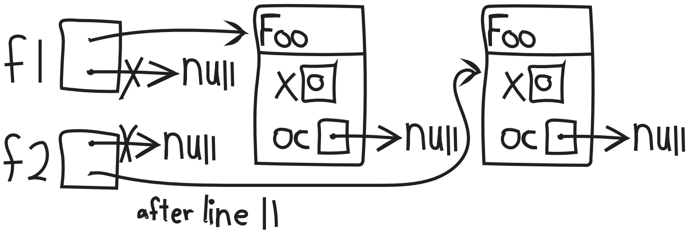


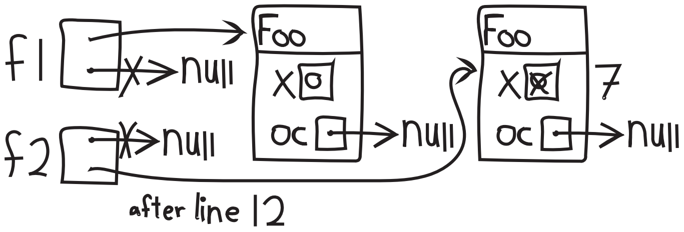
3. Using `new` to create an object instance of a class

Using Java's **new** creates an object instance of a class. Along with the box for the variable, you draw a rectangle for the object instance and at the top of the rectangle place the class's name. Then for each of the class's fields, add a box inside the object instance rectangle and label each box with the field's name. Then, use the code found in the class's constructor to initialize all these newly drawn fields. For object variable boxes, always draw the tail of the arrow in the variable's box and the head pointing to either *null* or the rectangle representing the object instance.

```
1   class Foo {
2     public int x;
3     public OtherClass oc;
4     public Foo() {
5        x = 0;
6        oc = null;
7     }
8   } // end Foo
9
10  Foo f1 = new Foo();
11  Foo f2 = new Foo();
12  f2.x = 7;
```



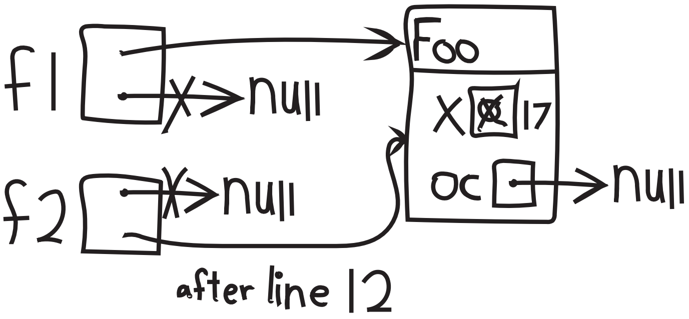




4. An assignment of a non-primitive type (object type)

If you see an assignment of a non-primitive type, then that copies a reference (i.e., in the diagram makes the variables point to the same object). The arrowhead of the assigned object points to whatever the original object pointed to.

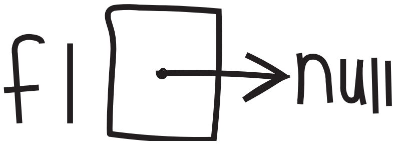
```
1    class Foo {
2       public int x;
3       public OtherClass oc;
4       public Foo() {
5          x = 0;
6          oc = null;
7       }
8    } // end Foo
9
10   Foo f1 = new Foo();
11   Foo f2 = f1;
12   f1.x = 17;
```



5. Using null

At the time of an object's declaration, the Java automatically initializes its reference to null.

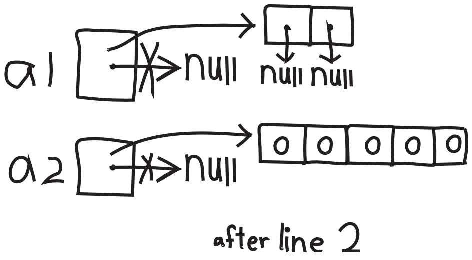
```
1    Foo f1;
```



6. Using `new` for a build-in Java array

A Java array is an object that contains a fixed-sized collection of other objects (see variable a1 below) or of primitive values (see a2 below). Draw the box to represent the array variable, then draw a grid to represent the array. Then make the array variable point to the array's grid. *Note that without a "new", no new arrays (grids) can be created.*

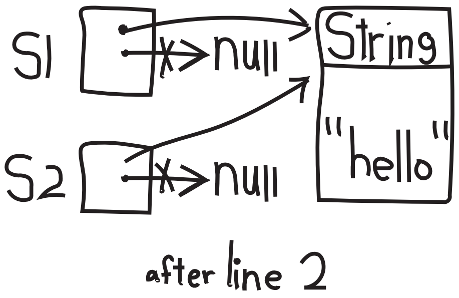
```
1    Foo[] a1 = new Foo[2];
2    int[] a2 = new int[5];
```



7. String Variables

Java String variables are objects Implemented by Java's String class. Draw a String variable and its contents like this.

```
1    String s1 = "hello";
2    String s2 = s1;
```


S1
null
String
S2
null
"hello"
after line 2