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.

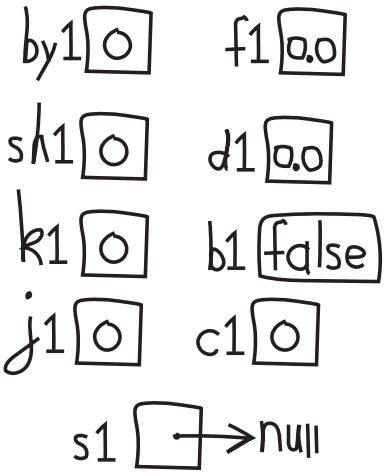
- Variable is a primitive type - then the box will contain the variable's actual value
- Variable is a non-primitive type, i.e., object - then the box will contain the tail of an arrow

2. Diagramming the default values for the 8 Java primitive types and for Java object types:

As Defined by Java	Example Declarations	Corresponding Box & Ptr Diagrams

Data Type	Default
byte	0
short	0
int	0
long	0L
float	0.0f
double	0.0d
boolean	false
char	'\u0000'
object (e.g., String)	null

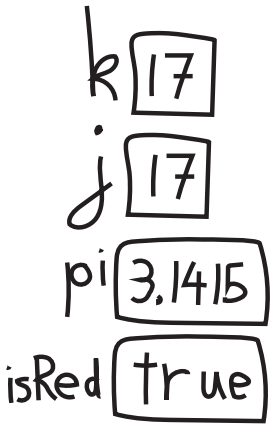
```
{
    byte by1;
    short sh1;
    int k1;
    long j1;
    float f1;
    double d1;
    boolean b1;
    char c1;
    String s1;
}
```



3. Assignment of an initial value to a primitive

If the code contains an assignment of a value to a primitive variable when the variable is declared, then place the value in the variable's box:

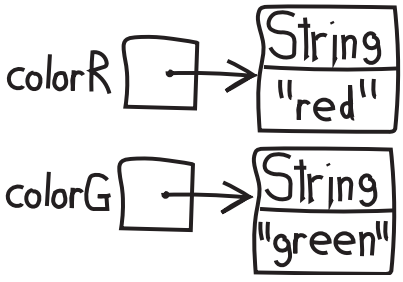
```
1   int k = 17;
2   int j = k;
3   double pi = 3.1415;
4   boolean isRed = true;
```



4. Assignment of initial value to an object reference

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, draw a line below the class's name, and then write in the initial value:

```
1   String colorR = "red";
2   String colorG = "green";
```

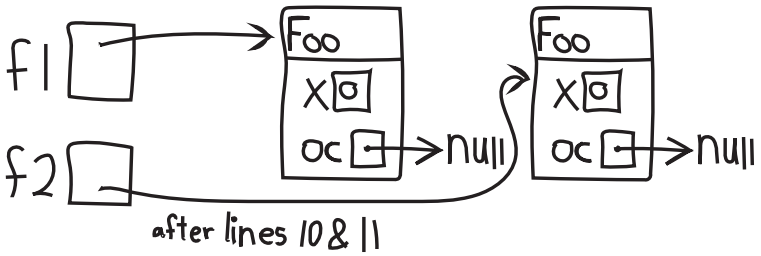


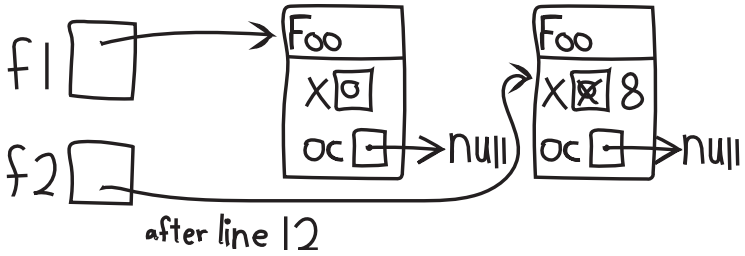
5. Using *new* to create an object instance of a class

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 a line under 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. Next, 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 = 8;
```

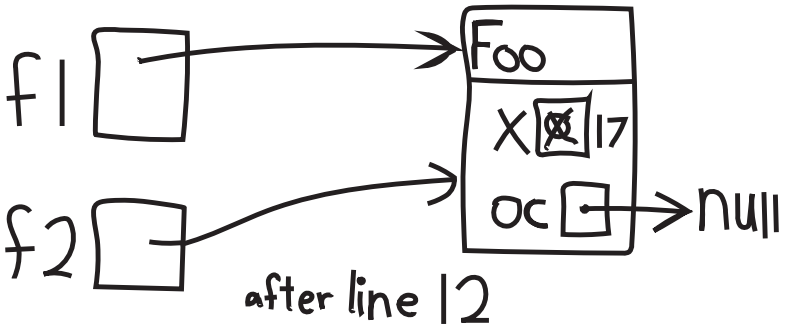




6. An assignment of a non-primitive type (object type)

If you see an assignment of a non-primitive type, that assignment copies a reference (see *line 11* below). The arrowhead of the assigned to variable points to whatever the original variable pointed to (i.e., in the diagram below, `f1` and `f2` point to the same 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 = f1;
12  f1.x = 17;
```



7. Using *new* for a built-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 can be created.*

```
1   Foo[] a1 = new Foo[2];
2   int[] a2 = new int[5];
3   a1[1] = new Foo();
4   a2[3] = 21;
```