

Since the discussion section had to be canceled because of snow we are asking you to complete the handout as a written homework assignment instead. This handout is due on Monday 09 February. Bring your completed answers on paper to hand in to your TA.

The goal of these exercises is to give you practice with references (also known as pointers) to objects and the problems newer Java programmers often have with them: aliasing, equality (`==`), and equivalence (`.equals()`).

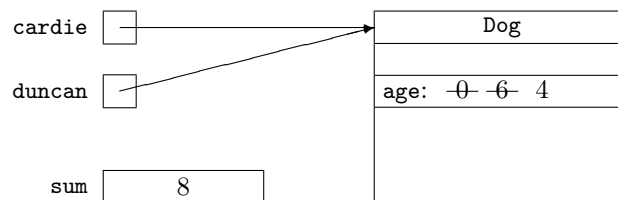
For full credit you must draw pictures so that we can be reasonably sure that you understand why you are giving the answer you are. For example, consider this problem from Lecture 2:

```

1 Dog cardie = new Dog();
2 Dog duncan = cardie;
3 cardie.setAge(6);
4 duncan.setAge(4);
5 int sum = cardie.getAge() + duncan.getAge();

```

You should end up with a picture that looks something like this:



References / Pointers

What does each expression evaluate to? (Remember your ABDs: Always Be Drawing.)

1.

```

1 String a = "foo";
2 String b = "bar";

```

(a) `a == b`

(b) `a.equals(b)`

2.

```

1 String c = "foo";
2 String d = c;

```

(a) `c == d`

(b) `c.equals(d)`

3.

```
1 String e = new String("foo");  
2 String f = new String("foo");
```

(a) `e == f`(b) `e.equals(f)`

4. For this and subsequent questions, you can assume getters and setters with the obvious names, and an `.equals()` method that compares the instance variable of each class for equivalence.

```
1 Person r = new Person("Abhyuday");  
2 Person s = new Person("Jagannatha");  
3 s.setName("Abhyuday");
```

(a) `r == s`(b) `r.equals(s)`

5.

```
1 Person t = new Person("Gene");  
2 Person u = t;  
3 u.setName("Keen");
```

(a) `t.getName()`(b) `t == u`(c) `t.equals(u)`

6.

```
1 Person v = new Person("Kris");  
2 Person w = v;  
3 w.setName("Kristina");  
4 v = new Person("Darlene")
```

(a) `v.getName()`(b) `w.getName()`(c) `w == v`(d) `w.equals(v)`

7.

```
1 Person x = new Person("Marc");  
2 Person y = x;  
3 y.setName("Marcus");  
4 x = new Person("John");  
5 x.setName("Marcus");
```

(a) `x.getName()`(b) `y.getName()`(c) `x == y`(d) `x.equals(y)`

8. Some of the code for linked lists we covered in lecture follows.

```
1 public class Chain {  
2     private int contents;  
3     private Chain next;  
4     public Chain(int contents, Chain next) {  
5         this.contents = contents;  
6         this.next = next;  
7     }  
8 }
```

(a) Suppose the following code executes.

```
1 Chain a = new Chain(34, null);  
2 Chain b = new Chain(42, a);  
3 Chain c = new Chain(69, b);
```

Draw the objects that are created, showing their `contents` and `next` references. Also show the variables `a`, `b`, and `c`, and the objects they refer to.

(b) Suppose the following code executes

```
1 Chain d = new Chain(69, new Chain(42, new Chain(34, null)));
```

i. Draw the created objects and references as before.

Assume a version of `equals()` (not shown) exists that checks whether two chains contain the same value(s) in `contents` in the same order.

ii. `a == d`

iii. `a.equals(d)`