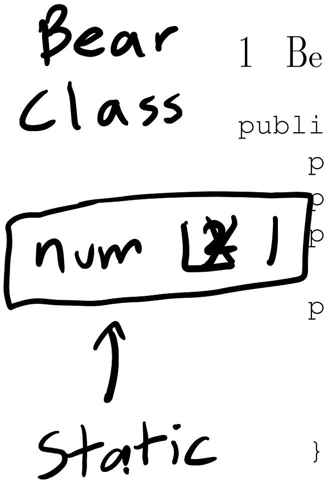
CS 61BL Summer 2021

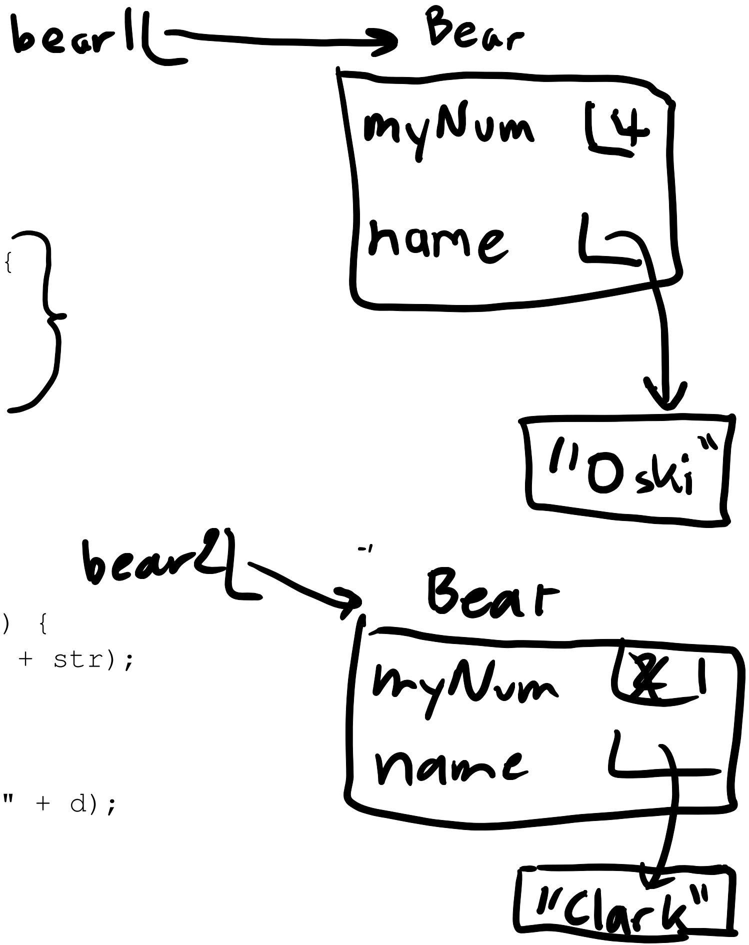
Small Group Tutoring Section 1

Worksheet 1

1 Bear



```
public class Bear {
    public static int num = 0;
    public int myNum;
    public String name;

    public Bear (int n, String str) {
        num += 1;
        myNum = n;
        name = str;
    }

    public void printNum() {
        System.out.println(myNum);
    }

    public void printInfo(String str) {
        System.out.println("I like " + str);
    }

    public void printInfo(int d) {
        System.out.println("Number: " + d);
    }
}
```



Take a look at the class and answer the question below. Suppose we instantiate the following two objects:

```
bear1 = new Bear(4, "Oski");
bear2 = new Bear(2, "Clark");
```

Determine the output after executing the following snippet of code.

```
System.out.println(bear2.num); 2
bear2.num -= 1;
System.out.println(bear1.num); 1
bear2.myNum -= 1;
System.out.println(bear1.myNum);4
bear1.printInfo(2);                        Number: 2
bear1.printInfo("apples");
                         I like apples
```

2 Box and Pointer Diagrams

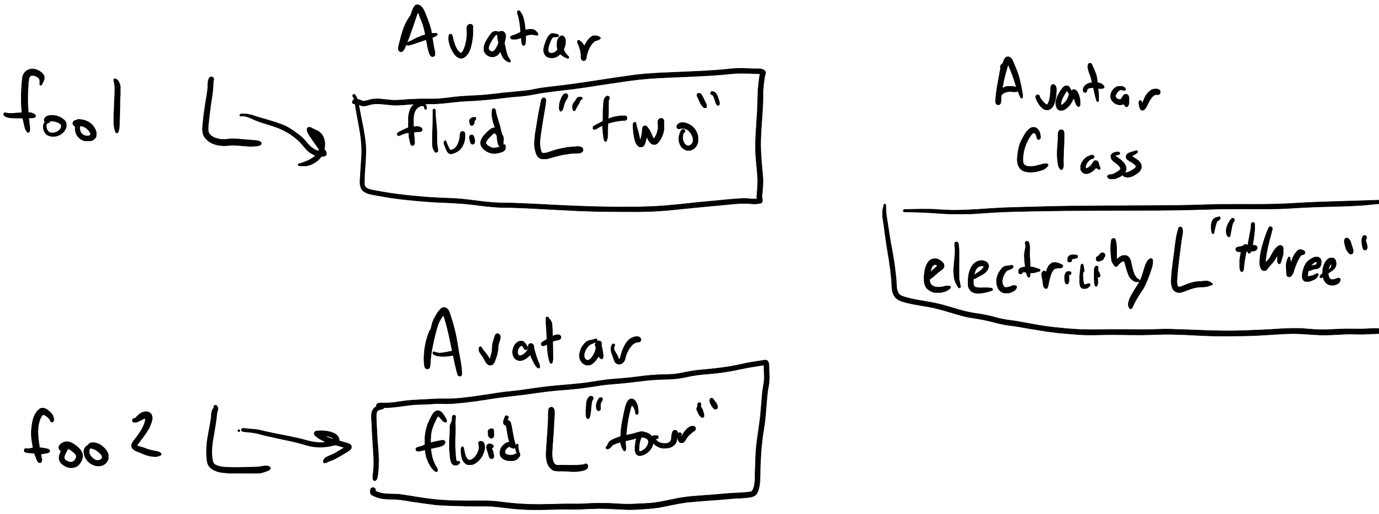
Answer the following questions about the `Avatar` class.

```
public class Avatar {
    public static String electricity; public String fluid;

    public Avatar(String str1, String str2) {
        Avatar.electricity = str1;
        this.fluid = str2;
    }

    public static void main(String[] args) {
        Avatar foo1 = new Avatar("one ", "two");
        Avatar foo2 = new Avatar("three ", "four");
        /* a */
        foo1.electricity = "I declare ";
        foo1.fluid = "a thumb war";
        /* b */
        foo2 = foo1;
        /* c */
    }
}
```

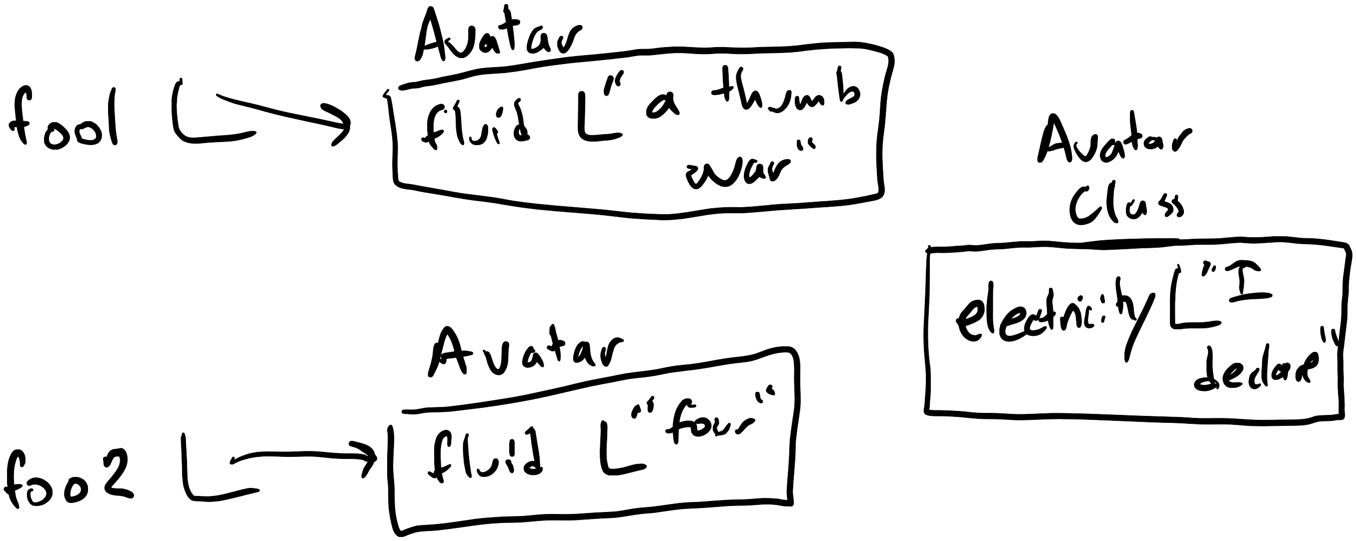
Draw the box-and-pointer diagrams of the states of the program during the lines with the comments a, b, and c in the main method before exiting.

On next page

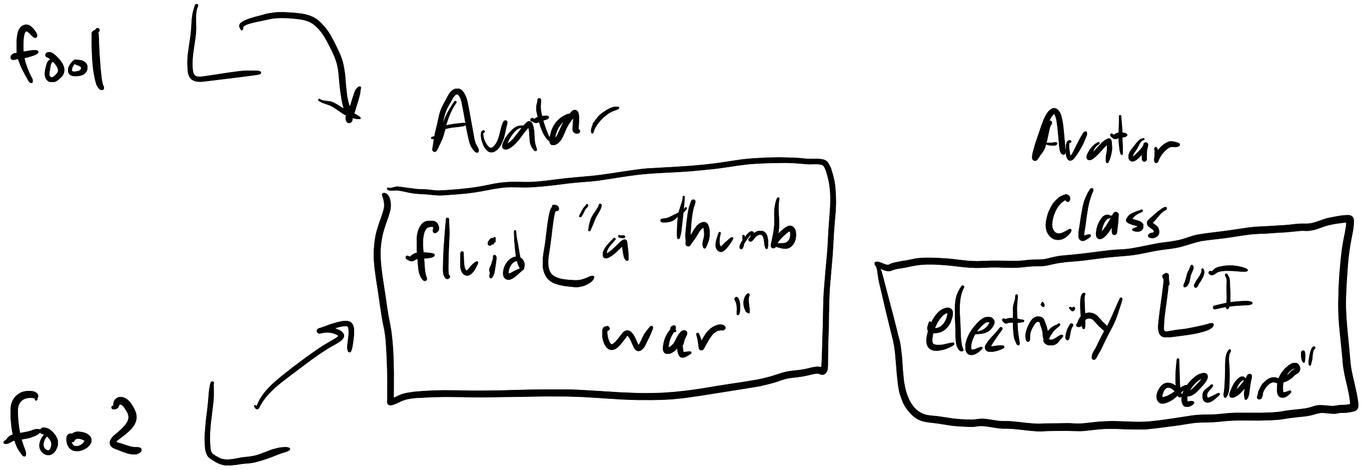
a)


Avatar
foo1
fluid "two"
Avatar Class
electricity "three"
Avatar
foo2
fluid "four"

b)


Avatar
foo1
fluid "a thumb war"
Avatar Class
electricity "I declare"
Avatar
foo2
fluid "four"

c)


foo1
Avatar
fluid "a thumb war"
Avatar Class
electricity "I declare"
foo2

3 Java Practice

1. Write a function that sums up all the digits in an integer recursively. For example, `sumDigits(31415)` should return $3+1+4+1+5=14$.

```
public static int sumDigits(int x) {
    if (x <= 0) {
        return 0;
    }
    return x % 10 + sumDigits(x / 10);
}
```

5 + (1 + (4 + (1 + (3 + (0 ⟶ ...

2. Extra - Write a function that sums up all the digits in an integer iteratively.

```
public static int sumDigits(int x) {
    int total = 0;
    while (x > 0) {
        total += x % 10;
        x /= 10;
    }
    return total;
}
```

⟷

```
int total = 0;
for (int a = x; a > 0; a /= 10) {
    total += a % 10;
}
return total;
```

5 ÷ 2 = 2 → Java int division

4 Do you Git it?

Recall the Git commands `init`, `add`, `commit`, `status`, `log`, `show`, `clone`, `pull`, and `push`. Circe the command which allows you to...

1. tell Git to start tracking a file:
   `init` [✓ `add`] `commit` `status` `log` `show` `clone` `pull` `push`

2. save a snapshot of the files being tracked:
   `init` `add` [✓ `commit`] `status` `log` `show` `clone` `pull` `push`

3. see what files have changed since your last commit:
   init add commit [✓ status] log show clone pull push

4. see a list of previous commits:
   init add commit status [✓ log] show clone pull push

5. create a local copy of a remote repository:
   init add commit status log show [✓ clone] pull push

6. send your locally tracked files to a remote repository:
   init add commit status log show clone pull [✓ push]