CS 61BL Summer 2019 — Lab 2

June 25, 2019

Name: Jon Notes

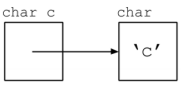
SID:

Write your name and login above. Please complete this worksheet during your lab, and turn it in to your TA by the end of your section. You are encouraged to work with your partners and neighbors collaboratively.

1 Drawing a char Variable

1.1 What's wrong with this box and pointer diagram for the code:

```
1 char c;
2 c = 'c';
```



should not be a reference

Char: c

int: 1

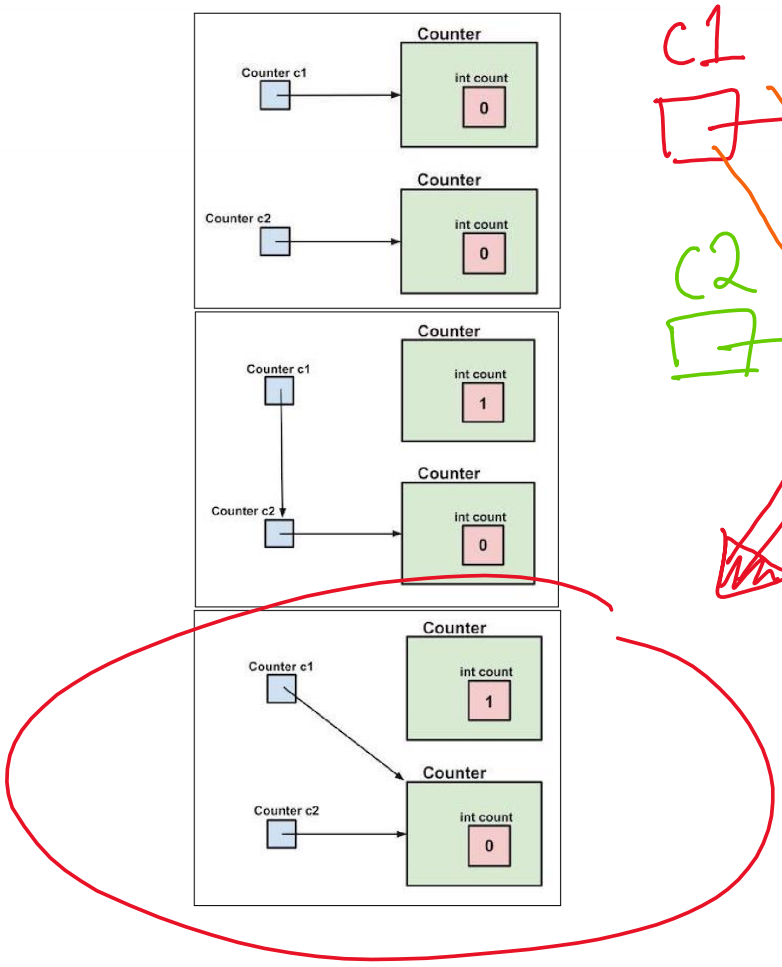
boolean: true

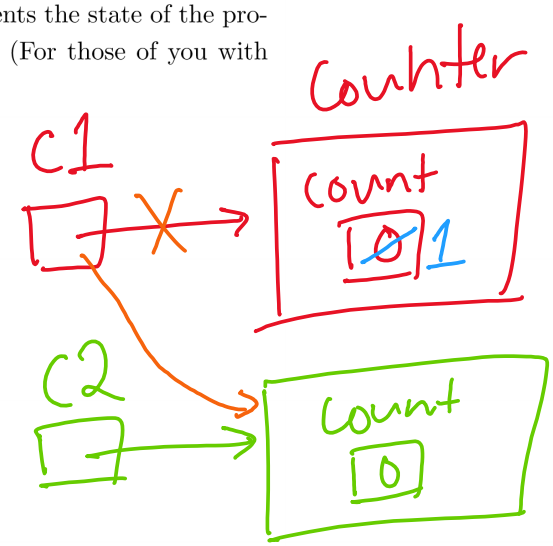
2 Counter

2.1 Consider a main program for the `Counter` class.

```
1  public class Counter {
2
3      int count = 0;
4
5      void increment() {
6          count = count + 1;
7      }
8
9      public static void main(String[] args) {
10         Counter c1 = new Counter();
11         c1.increment();
12         Counter c2 = new Counter();
13         c1 = c2;
14     }
15 }
```

Circle the box-and-pointer diagram which best represents the state of the program at the end of the main method before exiting. (For those of you with some Java-foo, there is no garbage collection)





3 Counter Problems

For each question in this exercise, choose a response from this list:

A. `c1` cannot be resolved.

B. `count` must be `private`.

C. Cannot make a static reference to the non-static method `increment()` from the type `Counter`.

D. The constructor `Counter(int)` is undefined.

E. The method `increment()` in the type `Counter` is not applicable for the arguments `(int)`.

F. Cannot make a static reference to the non-static field `count`.

3.1 Which letter response from above describes the problem with this `Counter` class?

```
1  public class Counter {
2
3      int count = 0;
4
5      void increment() {
6          count = count + 1;
7      }
8
9      public static void main (String[] args) {
10         Counter c1 = new Counter();
11         increment();
12         c1.count = 0;
13     }
14 }
```

static: not associated with the class, think of it as running "outside of" the class

increment() needs to be called from an instance!

⇒ C

different from before

3.2 Which letter response from above describes the problem with this `Counter` class?

```
1  public class Counter {
2
3      int count = 0;
4
5      void increment() {
6          count = count + 1;
7      }
8
9      public static void main (String[] args) {
10         Counter c1 = new Counter();
11         c1.increment();
12         count = 0;
13     }
14 }
```

(this is a field)

same as above, but now referencing a field instead of a method

⇒ F

3.3 Which letter response from above describes the problem with this `Counter` class?

```
1  public class Counter {
2
3      private int count = 0;
4
5      void increment () {
6          count = count + 1;
7      }
8
9      void setMyCount(int count) {
10         count = count;
11     }
12
13     public static void main(String [] args) {
14         Counter c1 = new Counter();
15         c1.increment(2);
16         c1.setMyCount(0);
17     }
18 }
```

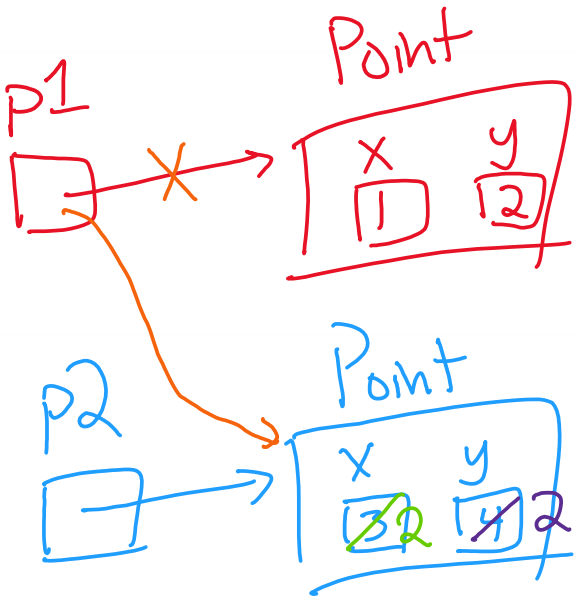
increment doesn't take arguments!

⇒ E

4 Assignment Statements

4.1 Draw a box and pointer diagram in order to tell me what gets printed by the following program.

```
1  import java.awt.Point;
2
3  public class Test {
4
5      public static void main(String[] args) {
6          Point p1 = new Point ();
7          p1.x = 1;
8          p1.y = 2;
9
10         Point p2 = new Point ();
11         p2.x = 3;
12         p2.y = 4;
13
14         // now the fun begins
15         p2.x = p1.y;
16         p1 = p2;
17         p1.y = p2.x;
18         System.out.println (p1.x + " " + p1.y + " " + p2.x + " " + p2.y);
19     }
20 }
```



5 Static Methods and Variables

```
1  public class Cat {
2     public String name;
3     public static String noise;
4
5     public Cat(String name, String noise) {
6         this.name = name;
7         this.noise = noise;
8     }
9
10    public void play() {
11       System.out.println(noise + " I'm " + name + " the cat!");
12    }
13
14    public static void anger() {
15       noise = noise.toUpperCase();
16    }
17    public static void calm() {
18       noise = noise.toLowerCase();
19    }
20 }
```

5.1 Draw a box and pointer diagram in order to determine what will happen after each call of `play()` in the following method.

```
1  public static void main(String[] args) {
2     Cat a = new Cat("Cream", "Meow!");
3     Cat b = new Cat("Tubbs", "Nyan!");
4     a.play();
5     b.play();
6     Cat.anger();
7     a.calm();
8     a.play();
9     b.play();
10 }
```