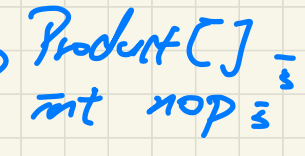
Lecture 4 - Sep. 17

Review of OOP

Reference Aliasing & Primitive Arrays
Ref-Typed Parameters vs. Return Values

Announcements/Reminders

- Lab0P2 due this Friday at 12 noon!
- Lab1 to be released after Lab0P2 is due.
- Priority: Lab0P2 (tutorial videos + PDFs)
- This Thursday's office hour will be re-scheduled.



Copying Primitive Values

```
int i1 = 1;
int i2 = 2;
int i3 = 3;
int[] numbers1 = {i1, i2, i3};
int[] numbers2 = new int[numbers1.length];
for(int i = 0; i < numbers1.length; i ++) {
numbers2[i] = numbers1[i];
}
numbers1[0] = 4;
System.out.println(numbers1[0]);
System.out.println(numbers2[0]);
```

0 1 2

3

1st it. i == 0

n2[0] = n1[0];

2nd it. i == 1

n2[1] = n1[1];

3rd it. i == 2

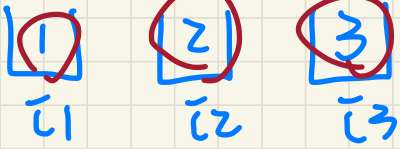
n2[2] = n1[2]

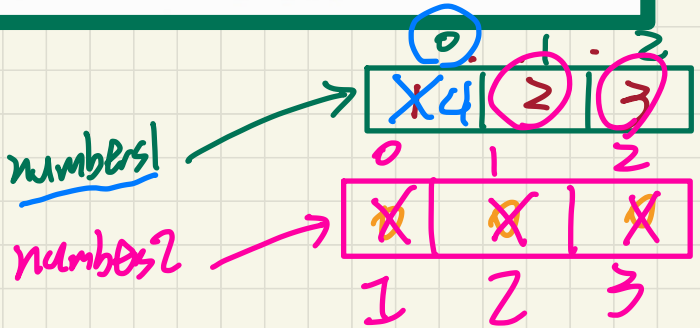
after i++

i == 3

i < n1.length

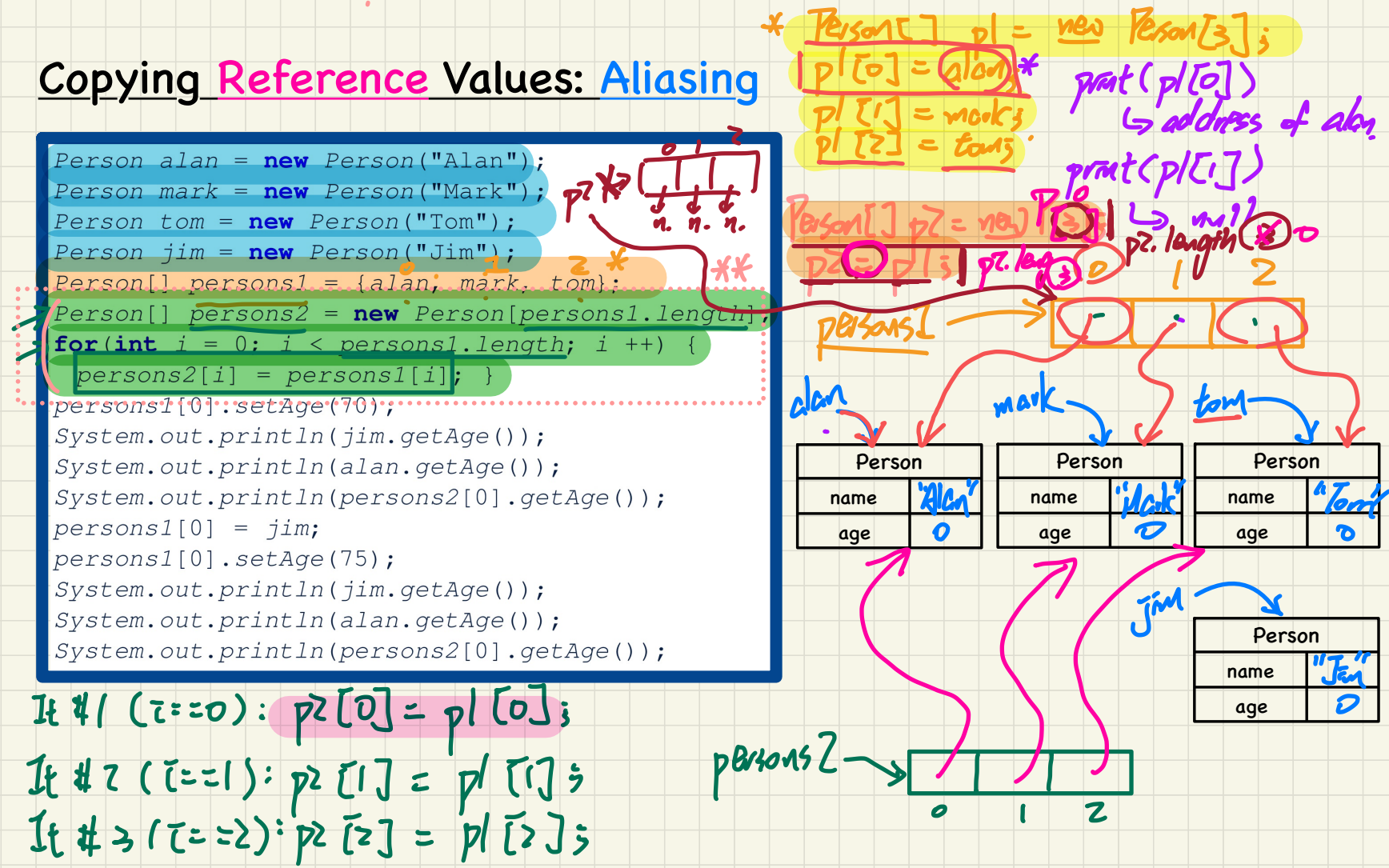
F





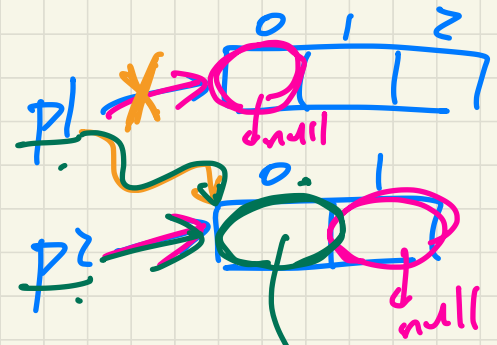
Copying Reference Values: Aliasing

```
Person alan = new Person("Alan");
Person mark = new Person("Mark");
Person tom = new Person("Tom");
Person jim = new Person("Jim");
Person[] persons1 = {alan, mark, tom};
Person[] persons2 = new Person[persons1.length];
for(int i = 0; i < persons1.length; i ++) {
  persons2[i] = persons1[i]; }
persons1[0].setAge(70);
System.out.println(jim.getAge());
System.out.println(alan.getAge());
System.out.println(persons2[0].getAge());
persons1[0] = jim;
persons1[0].setAge(75);
System.out.println(jim.getAge());
System.out.println(alan.getAge());
System.out.println(persons2[0].getAge());
```



```
Person p1 = new Person[3];
Person p2 = new Person[2];
```

jvm → Person



p1.length 3

p2.length 2

p1 == p2 false

p1[0] == p2[1] true

null null

p1[0] == jvm ✓ p1[0] == p2[0] ✓

p2[0] == jvm

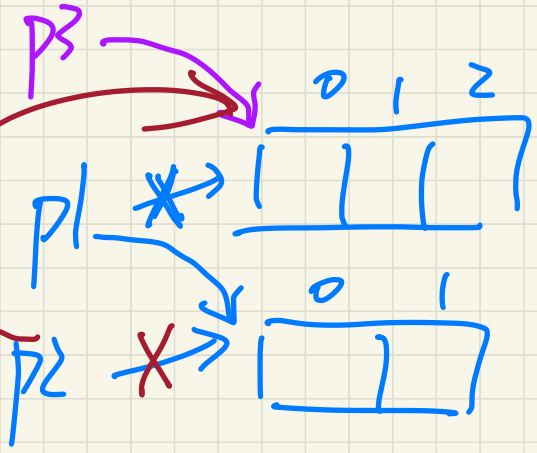
aliasing (multiple variables store same address)

p1 = p2;

p1[0] = jvm;

p1 == p2

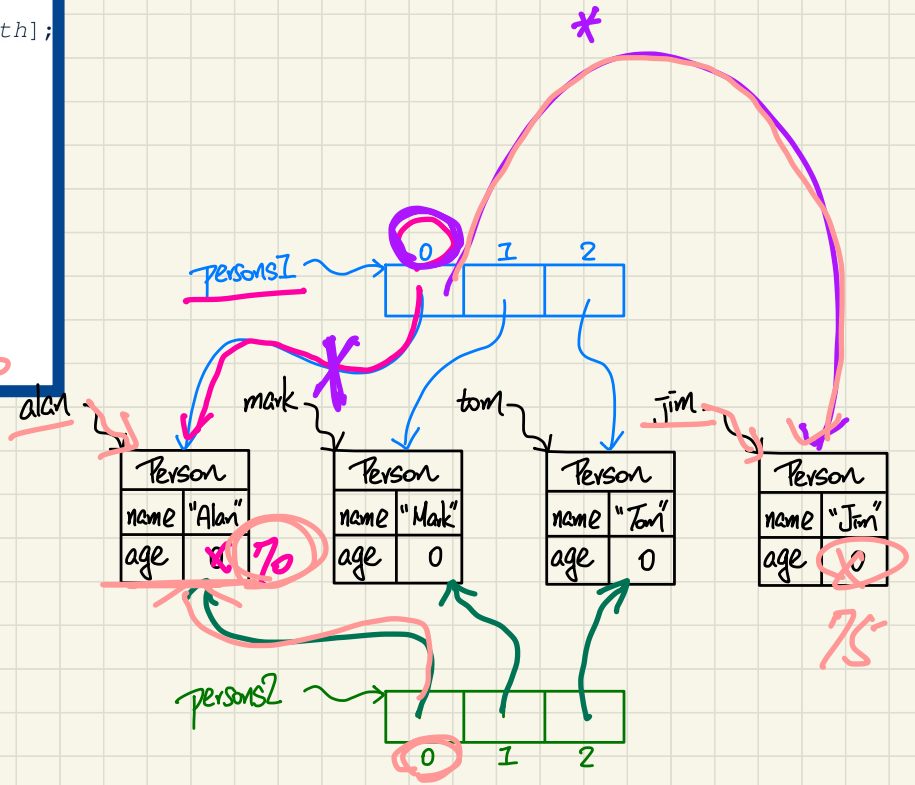
```
Person[] p1;
Person[] p2;
Person[] p3;
```




```
p1 = new Person[3];
p2 = new Person[2];
p3 = p1;
p1 = p2;
```

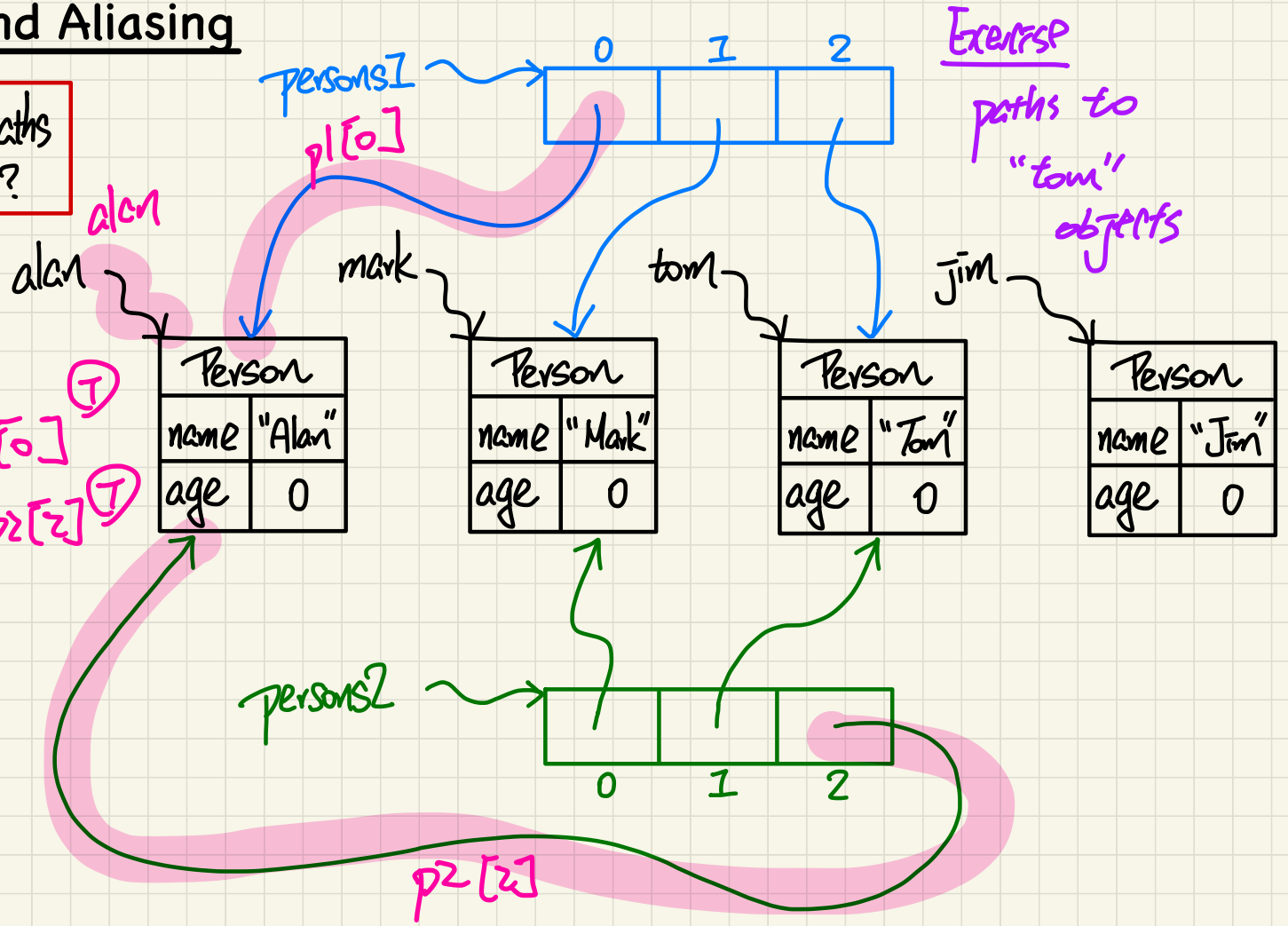
```
p2 = p3;
```

```java
Person alan = new Person("Alan");
Person mark = new Person("Mark");
Person tom = new Person("Tom");
Person jim = new Person("Jim");
Person[] persons1 = {alan, mark, tom};
Person[] persons2 = new Person[persons1.length];
for(int i = 0; i < persons1.length; i ++) {
  persons2[i] = persons1[i]; }
persons1[0].setAge(70);
System.out.println(jim.getAge());
System.out.println(alan.getAge());
System.out.println(persons2[0].getAge());
persons1[0] = jim;
persons1[0].setAge(75);
System.out.println(jim.getAge());
System.out.println(alan.getAge());
System.out.println(persons2[0].getAge());
```



Arrays and Aliasing

All alias paths to "Alan"?



Exercise

paths to "tom" objects

Reference-Typed Return Values


```
public class Point {
   /* A mutator modifying the context Point object */
   public void moveUp (int y) {
      this.y = this.y + y;
   }

   /* An accessor returning a new Point object */
   public Point movedUpBy(int y) {
      Point np = new Point(this.x, this.y);
      np.moveUp(y);
      return np;
   }
}
```

```
public class PointTester {
   public static void main(String[] args) {
      Point p1 = new Point(2.5, -3.6);
      p1.moveUp(7.8);
      Point p2 = p1.movedUpBy(6.4);
      System.out.println(p1 == p2);
   }
}
```