Lecture 5 - Sep. 19

Review of OOP

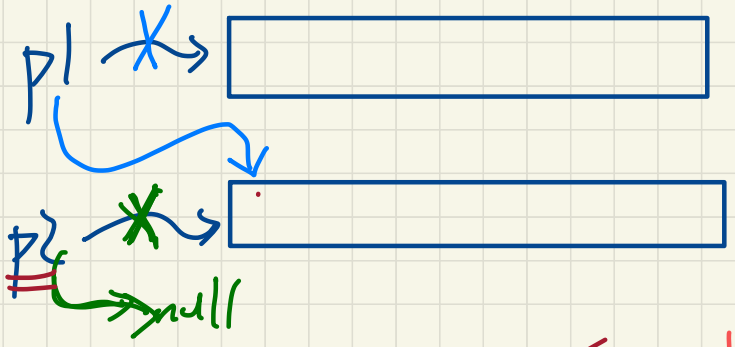
Ref-Typed Parameters vs. Return Values
Anonymous Objects
Reference to this
Static Variables

Announcements/Reminders

- LabOP2 due tomorrow (Friday) at 12 **noon**!
- Lab1 to be released after LabOP2 is due.
- In-Lab demo on the **Programming Pattern**
- Today's office hour to be re-scheduled
- **Mockup Programming Test** next Fri (5pm or 6pm)

Person[] p1 = new ____ ...

→ Person[] p2 = new ____ ..



p1 = p2 ;

p2 = null ;

① p1 == null (F)

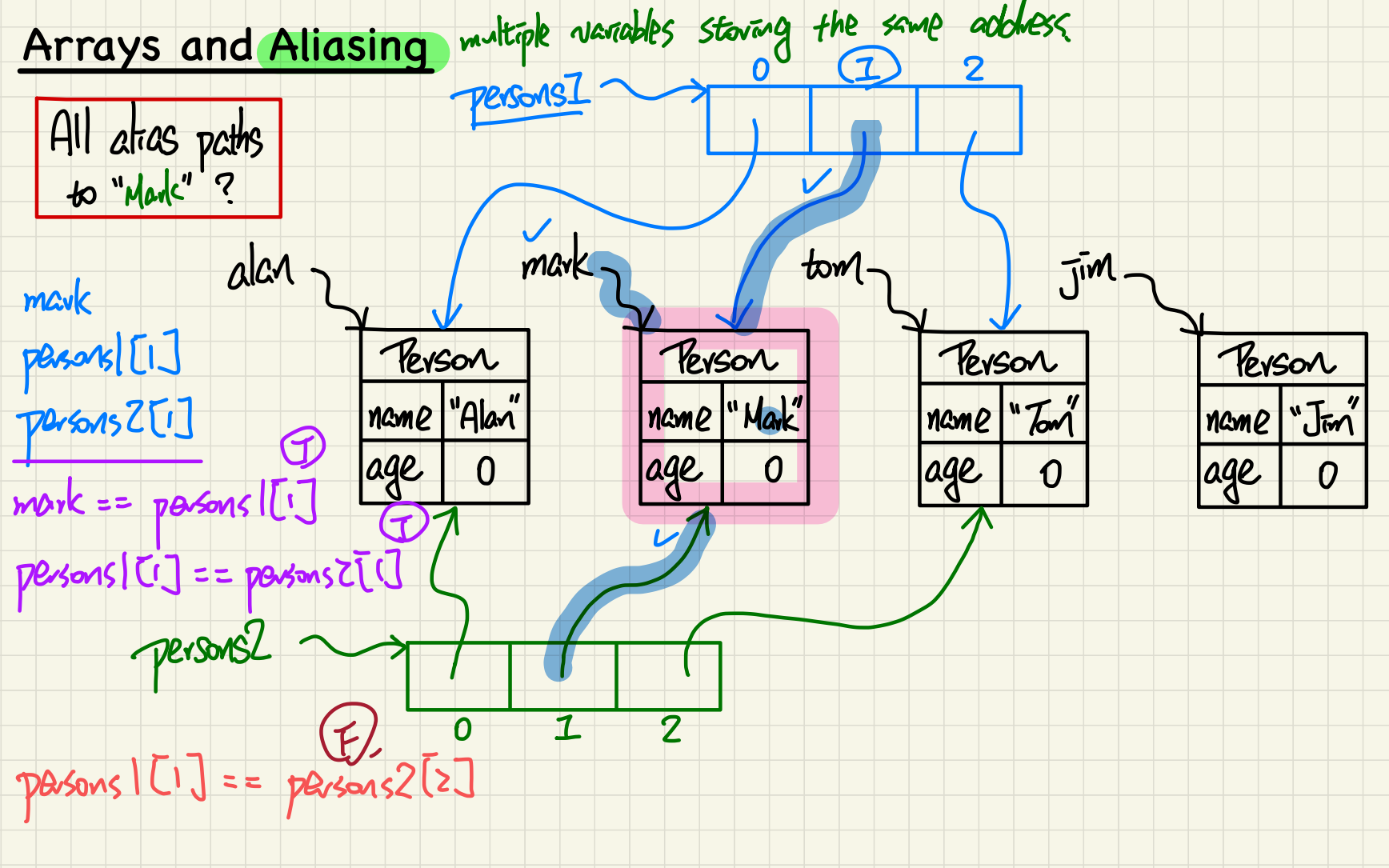
② p2 == null (T)

③ p1 == p2 (F)

Arrays and Aliasing

multiple variables storing the same address

All alias paths to "Mark"?

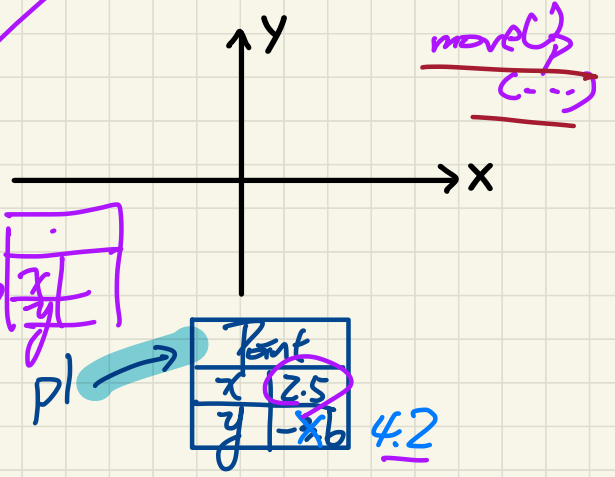


Reference-Typed Return Values


```
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 x) {
    Point np = new Point(this.x, this.y);
    np.moveUp(x);
    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);
  }
}
```



helper method call.

Point p3 = p1.movedUpBy(6.4); returns the address of some Point object

Anonymous Objects


```
1 double square(double x) {
2   double sqr = x * x;
3   return sqr; }
```

anonymous exp.

```
1 double square(double x) {
2   return x * x; }
```

```
1 Person getP(String n) {
2   Person p = new Person(n);
3   return p; }
```

anonymous obj

```
1 Person getP(String n) {
2   return new Person(n); }
```

Exercise

```
class Member {
  private Order[] orders;
  private int noo;
  /* constructor ommitted */
  public void addOrder(Order o) {
    this.orders[this.noo] = o;
    this.noo++;
  }
  public void addOrder(String n, double p, double q) {
    // 1. Order o = new Order(n, p, q);
    //    this.orders[noo] = o;
    //    this.noo++;
    // 2. this.addOrder(o);
    // 3. this.addOrder(_____);
  }
}
```

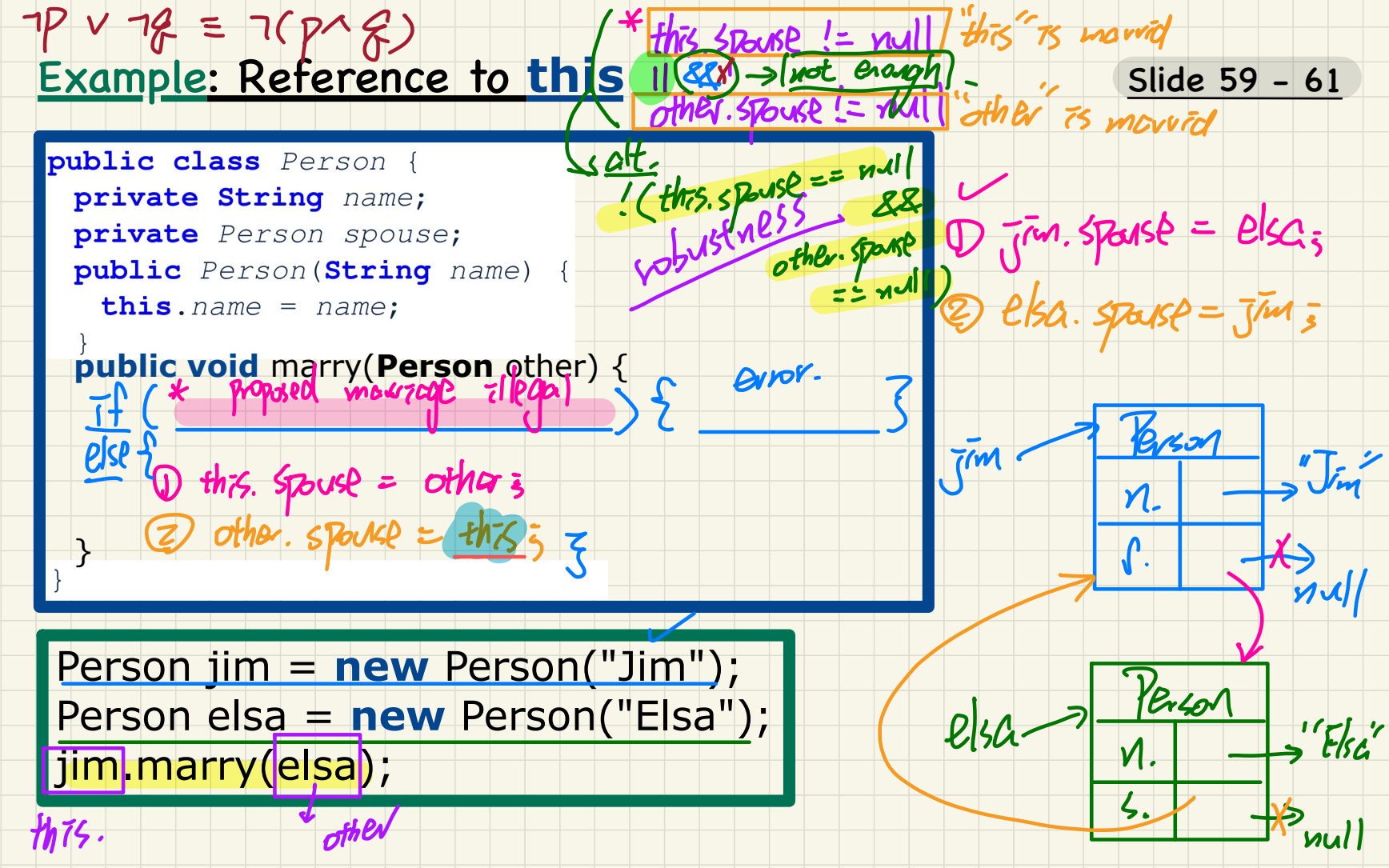
overloaded method

current class to look for the method called

Example: Reference to this

```
public class Person {
  private String name;
  private Person spouse;
  public Person(String name) {
    this.name = name;
  }
  public void marry(Person other) {
    if ( * proposed marriage illegal ) { error. }
    else {
      ① this.spouse = other;
      ② other.spouse = this;
    }
  }
}
```

```
Person jim = new Person("Jim");
Person elsa = new Person("Elsa");
jim.marry(elsa);
```



Exercise

```
void marry (Person other) {
    if ( ________________ ? ) {
        this.spouse = other;
        other.spouse = this;
    }
    else { error }.
}
```