

Lecture 4 - Sep. 17

Review of OOP

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

Announcements/Reminders

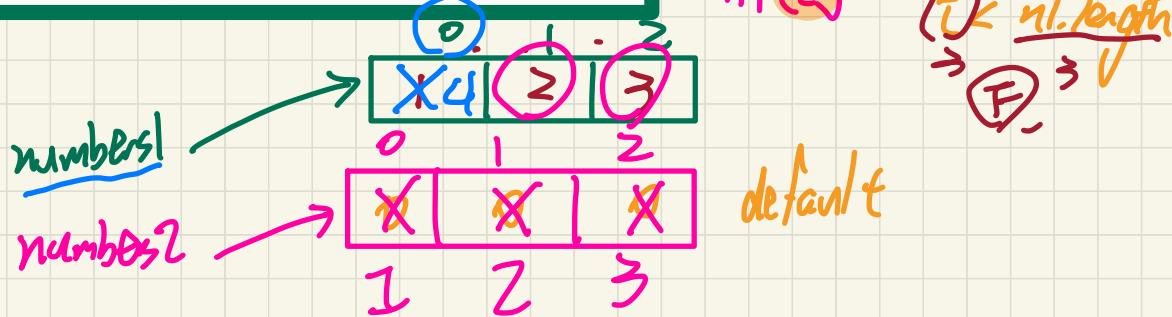
- LabOP2 due this Friday at 12 noon!
- Lab1 to be released after LabOP2 is due.
- Priority: LabOP2 (tutorial videos + PDFs)
- This Thursday's office hour will be re-scheduled.

Product [] -
int nOp =

Copying Primitive Values

```
int i1 = 1;  
int i2 = 2;  
int i3 = 3;  
int[] numbers1 = {i1, i2, i3};      0    1    2  
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]);
```

i1
i2
i3



Copying Reference Values: Aliasing

```

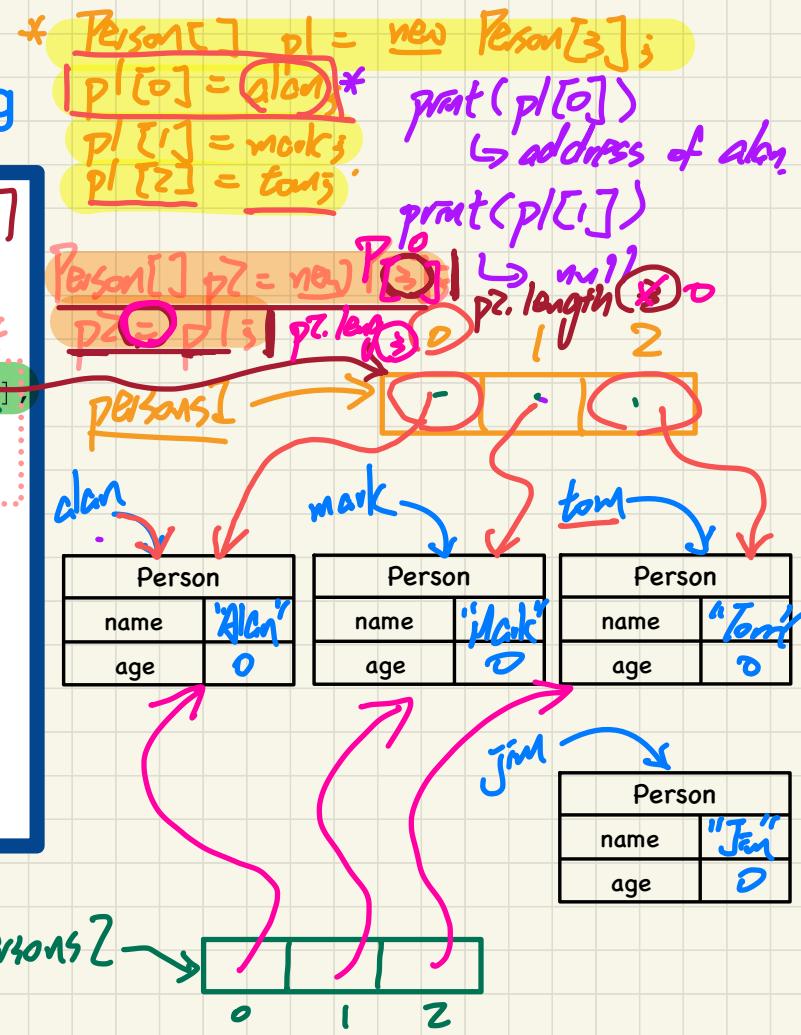
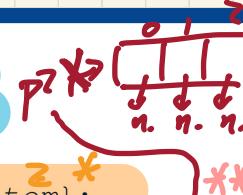
Person alan = new Person("Alan");
Person mark = new Person("Mark");
Person tom = new Person("Tom");
Person jim = new Person("Jim");
Person[] persons1 = {alan, mark, tom};
Person[] persons2 = new Person[persons1.length];
for(int i = 0; i < persons1.length; i++) {
    persons2[i] = persons1[i];
}
persons1[0].setAge(70);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

```

If #1 ($i=0$): $p2[0] = p1[0]$;

If #2 ($i=1$): $p2[1] = p1[1]$;

If #3 ($i=2$): $p2[2] = p1[2]$;



Person pl = new Person[β];

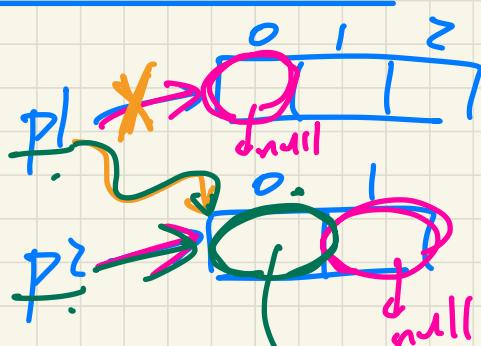
Person p2 = new Person[2];

A hand-drawn diagram on grid paper. On the left, the word "Jrun" is written in green. An arrow points from "Jrun" to a rectangular box on the right. Inside the box, the word "Person" is written vertically in green. A horizontal line extends from the bottom of the box to the right.

pl. length 3

p2. length 2

$p1 == pz$ false



$$\underline{p1[0]} == \underline{p2[1]}$$

$$P_1 = P_2$$

$\text{tmp} = \text{int} \sqrt{P[0]} = P[0]$

$\text{P}[0] = \text{V1}$
 $\text{P}[1] = \text{V2}$
 $\Rightarrow P[2]$

aliasing
(multiple
start
semp
address)

variables

Person [] p1 ;

Person [] p2 ;

Person [] p3 ;

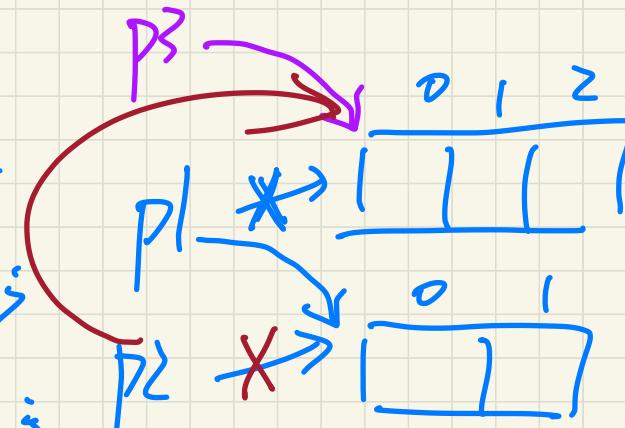
p1 = new Person [3] ;

p2 = new Person [2] ;

p3 = p1 ;

\Rightarrow p1 = p2 ; |

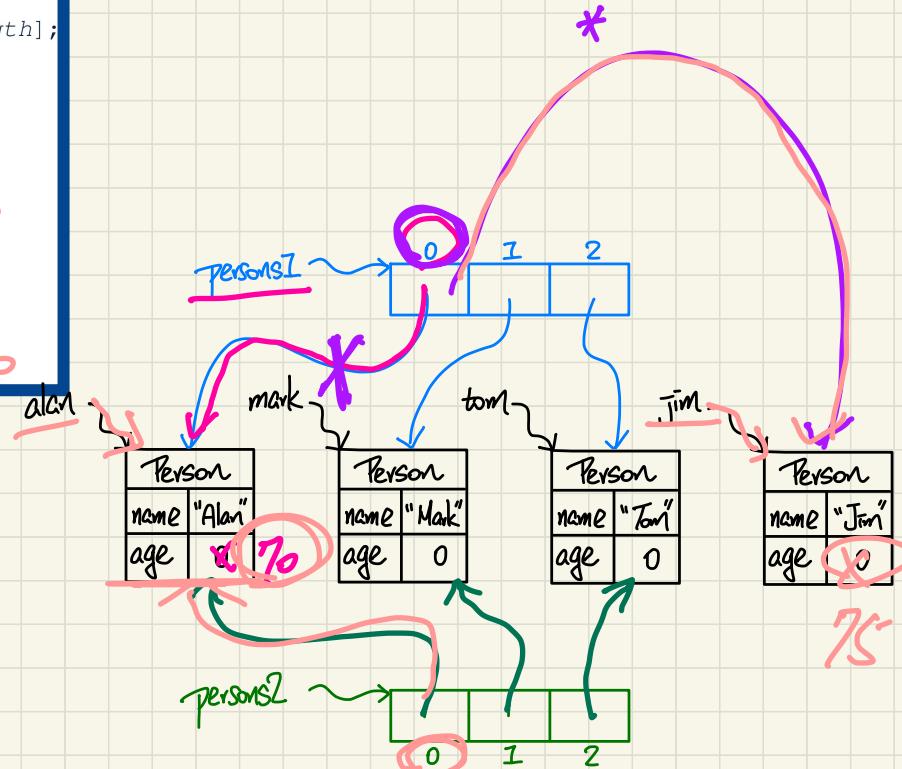
p2 = p3 ;



```

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()); 0 70
System.out.println(alan.getAge()); 70
System.out.println(persons2[0].getAge()); 70
persons1[0] = jim;
persons1[0].setAge(75); 75 70
System.out.println(jim.getAge()); 70
System.out.println(alan.getAge()); 70
System.out.println(persons2[0].getAge()); 70

```



Arrays and Aliasing

All alias paths
to "Alan"?

persons1

p1[0]

alan

0

1

2

Exercise

paths to
"Tom"
objects

alan == p1[0] (T)
p1[0] == p2[2] (T)

Person	
name	"Alan"
age	0

Person	
name	"Mark"
age	0

Person	
name	"Tom"
age	0

Person	
name	"Jim"
age	0

persons2

0	1	2

p2[2]

Reference-Typed Return Values

Slide 53

```
public class Point {  
    /* A mutator modifying the context Point object */  
    public void moveUp (int x) {  
        this.y = this.y + x; 7.8 7.8  
    } p1  
    /* An accessor returning a new Point object */  
    public Point movedUpBy(int y) {  
        Point np = new Point(this.x, this.y);  
        np.moveUp(x); 6.4 p1 p1  
        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);  
    }  
}
```

