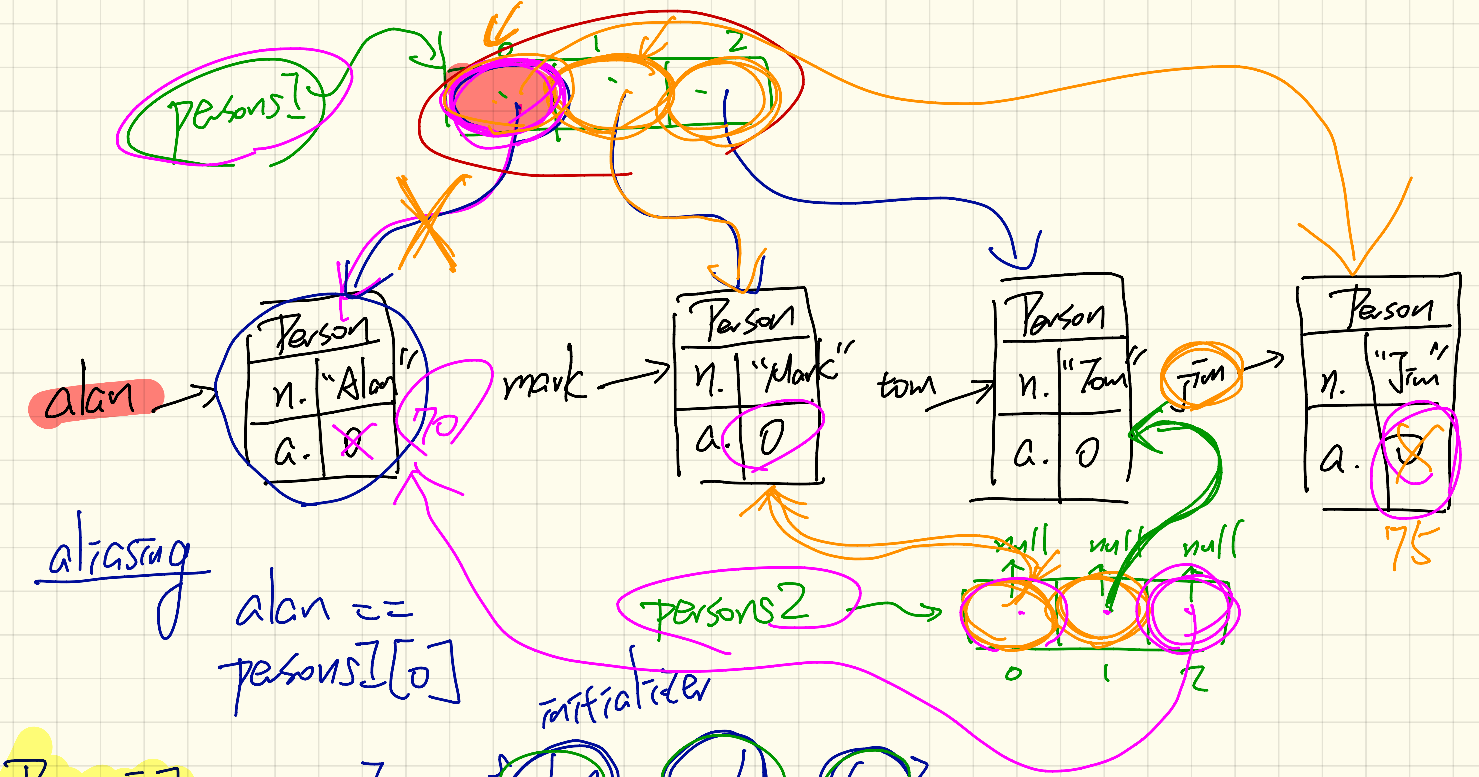
Wednesday Sep. 12

Lecture 3



aliasing

alan == persons1[0]

initializer

Person[] persons1 = {alan, mark, tom};

↳ Person[] persons1 = new Person[3];

persons1[0] = alan; ____[1] = mark; ____[2] = tom;

187 / 3.0

x / (double) y

```
for ( int i = 0; i < persons1.length; i++ ){
    persons2[i] = persons1[(i+1) % persons1.length];
}
```

3

0 1 2

1 2 0

3

i	(i+1) % 3
0	1
1	2
2	0

$187 / 3$ 62

$187 \% 3$ 1

x y integers

$$\underbrace{(x/y)}_{q} * y + (x \% y) = x$$

$$\underbrace{(187/3)}_{q} * 3 + \underbrace{(187 \% 3)}_{r} = 187$$

```
Person p = new Person("Jim");
```

store address of anynomaus object

v3

```
Order o = new Order(n, p, q);
addOrder(o);
```

v4

```
addOrder( new Order(n, p, q))
```

v1

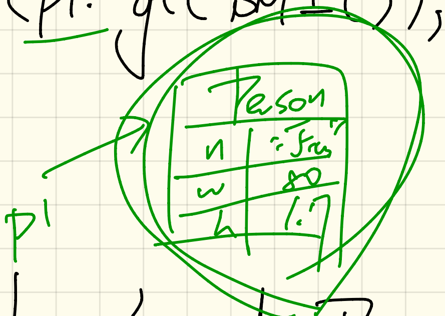
```
addOrder ( n, p, q ) {
    v2
    Order o = new Order(n, p, q);
    orders[noo] = o;
    noo++;
}
```

```
orders[noo] =
new Order(n, p, q)
;
```

If you only have to use the reference variable once, use anynomous object.

m1(){ L

```
Person p1 = new Person("Jim", 80, 1.7);
print(p1.getBMI());
print(p1.getBMI());
:
:
3
```

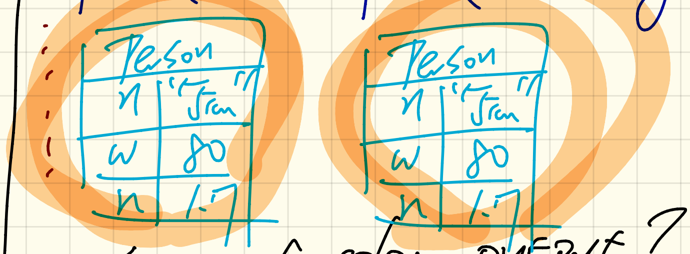


o.m1();
p1?

R

on the fly

```
1 print((new Person("Jim", 80, 1.7)).getBMI());
2 print((new Person("Jim", 80, 1.7)).getBMI());
```



Q1. L and R produce same console output ?

Q2. L and R have the same meaning ?
(visualization)

int p.t.

Integer r.t.

Wrapper

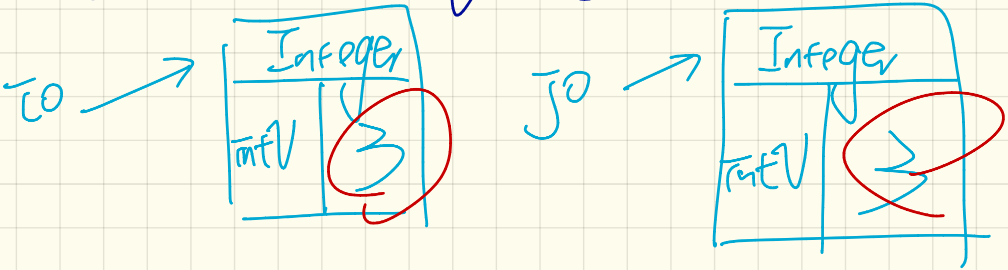
i == j T

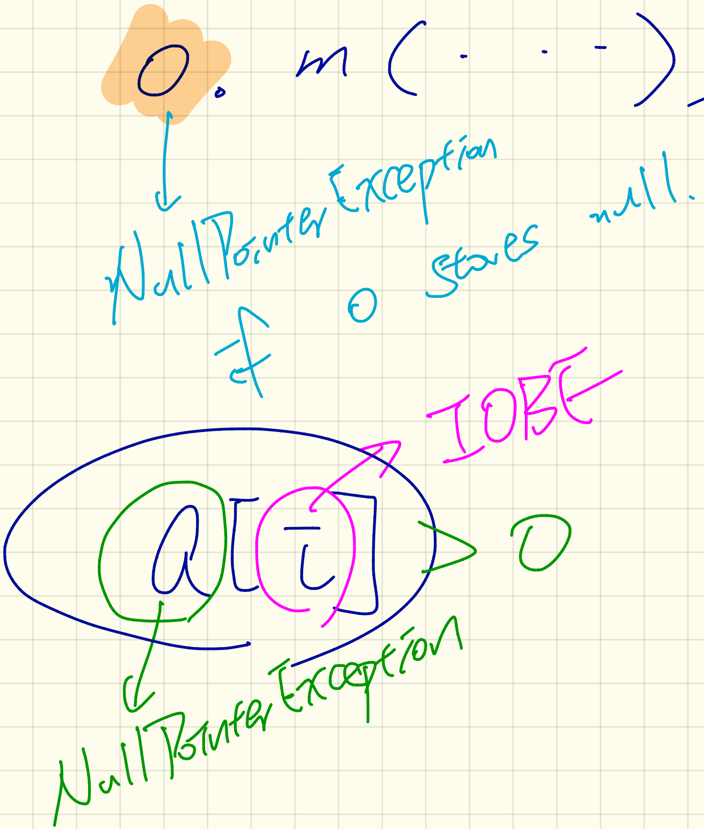
i0 == j0 F

i0.equals(j0) T

```
int i = 3;
int j = 3;

Integer i0 = new Integer(i);
Integer j0 = new Integer(j);
```




o.m(. . .);
NullPointerException
if o stores null.
a[i]
IOBE
> 0
NullPointerException

①　a[i] > 0

②　(0<=i && i<a.length)

③　(a!=null)

F

Q1. ③ && ② && ①

Q2. ② && ① && ③

a null

if a is null

NullPointerException when evaluating a.length

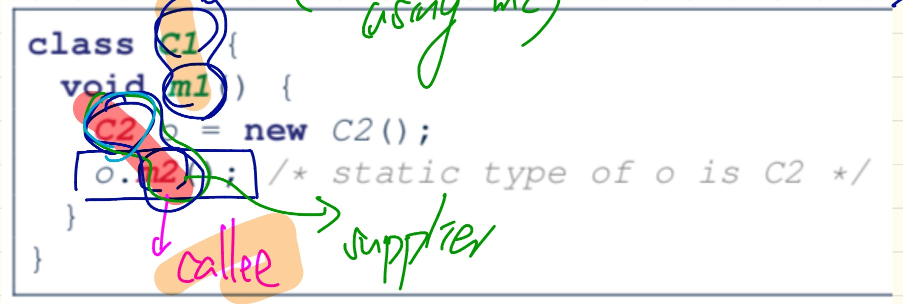
Caller vs. Callee

caller

client

supplier

caller (client using m2)



```
class C1 {
  void m1() {
    C2 o = new C2();
    o.m2(); /* static type of o is C2 */
  }
}
```

supplier

callee

Q. Can a method be a caller and a callee simultaneously?

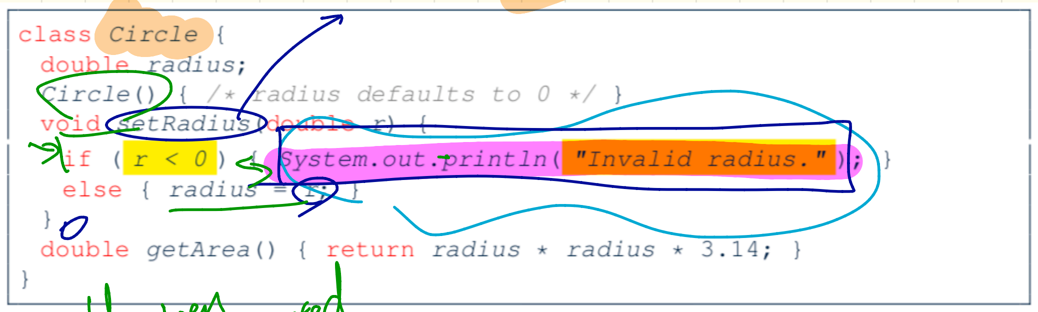
caller

```
class C2 {
  void m2() {
    C3 o = new C3();
    o.m3();
  }
}
```

callee

Error Handling with Console Messages: Circles

caller or callee?

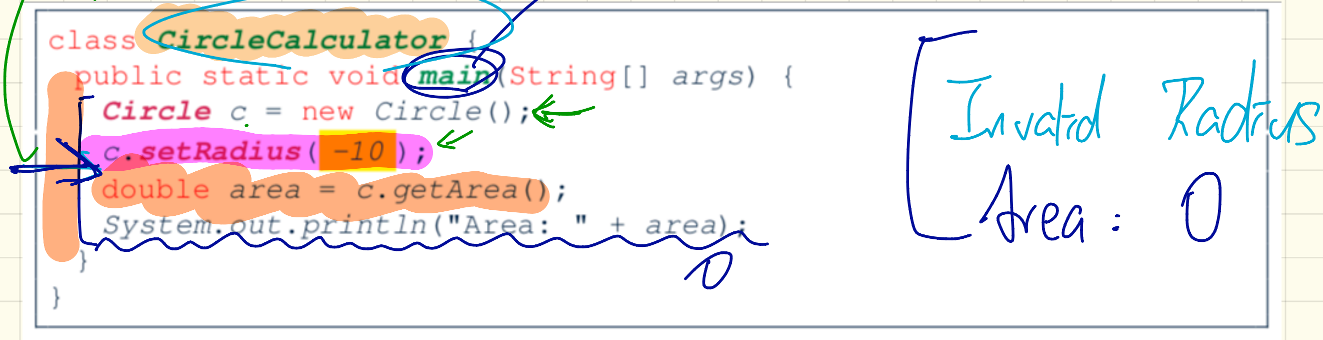


```
class Circle {
  double radius;
  Circle() { /* radius defaults to 0 */ }
  void setRadius(double r) {
   if (r < 0) { System.out.println("Invalid radius."); }
    else { radius = r; }
  }
  double getArea() { return radius * radius * 3.14; }
}
```

0

should have been interrupted.

caller or callee?



```
class CircleCalculator {
  public static void main(String[] args) {
   Circle c = new Circle();
   c.setRadius(-10);
   double area = c.getArea();
   System.out.println("Area: " + area);
  }
}
```

0

```
Invalid Radius.
Area: 0
```