

Lecture 6

Tuesday Sept. 26

```

class Point {
    double x;
    double y;

```

```

    void moveUp(int u) {

```

$p1$ this.y + = u; $2 \rightarrow 2$
 $p2$

```

    Point(double x, double y) {

```

$p2$ this.x = ~~x~~; $3 \rightarrow 6$
 $p2$ this.y = ~~y~~; $4 \rightarrow 8$
 $p1$

```

class PointTester {

```

```

    main() {

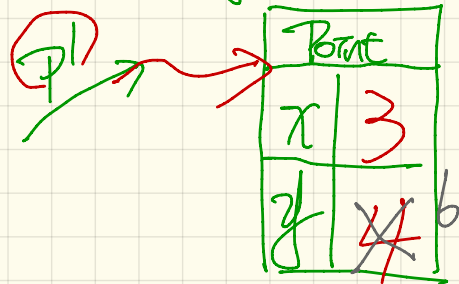
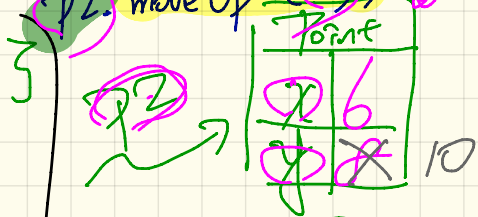
```

$Point$ $p1 = new$ $Point(3, 4)$;
 obj PCTF

$Point$ $p2 = new$ $Point(6, 8)$;
 obj PCTF

$p1$.moveUp(2);

$p2$.moveUp(2);



Print p1 = (new) Point (3, 4);

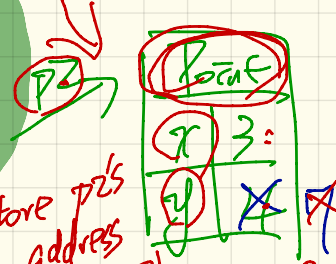
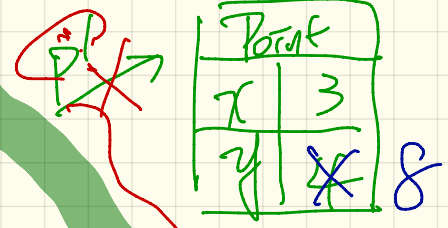
Print p2 = (new) Point (3, 4);

print(p1.x) print(p1.y) p2.x p2.y
3 4 3 4

p1.moveUp(4);

p2.moveUp(3);

p1.x p1.y p2.x p2.y
3 8 3 7



Store p2's address into p1
p1 = p2

p2.moveUp(2) 12

p1.moveUp(3)

p1.x p1.y p2.x p2.y
3 7 3 12
✓ ✗₁₂ ✓ ✓

Point p1 = new Point(3, 4);

Point p2 = new Point(6, 8);

p1 = p2;

p2 = p1;

p2.moveUp(3);

p1.moveUp(2);

p1.x == p2.x
p1.y == p2.y
10

Point	
x	3
y	4

p1

p1.x == p2.x
p1.y == p2.y
13

Point	
x	6
y	8

```
class Point {  
    double x;  
    double y;  
}
```

Point p = new Point(5, 3);

wrong
shadowing

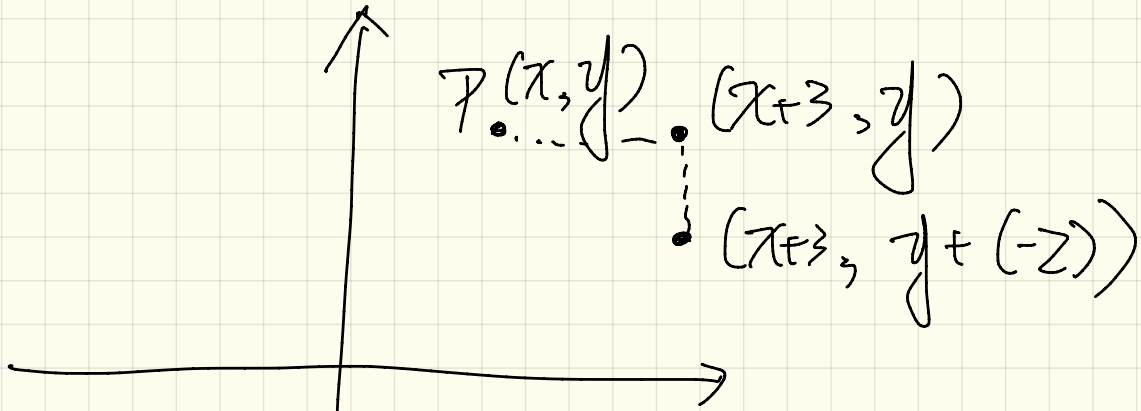
```
Point (double x, double y) {
```

? x = x;
y = y;

}

right

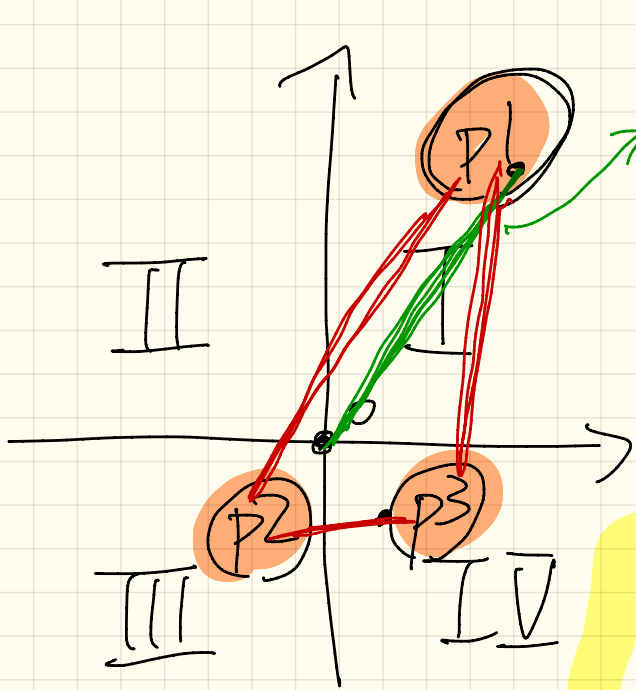
```
Point (double x, double y) {  
    this.x = x;  
    this.y = y;  
}
```



P. move $(0, 3, 0)$

\textcircled{P} . move $\left(\frac{3}{\downarrow}, \frac{-2}{\downarrow} \right)$

horizontal vertical



$p1.getV(p2)$

$p1.isAtFirstQuadrant()$

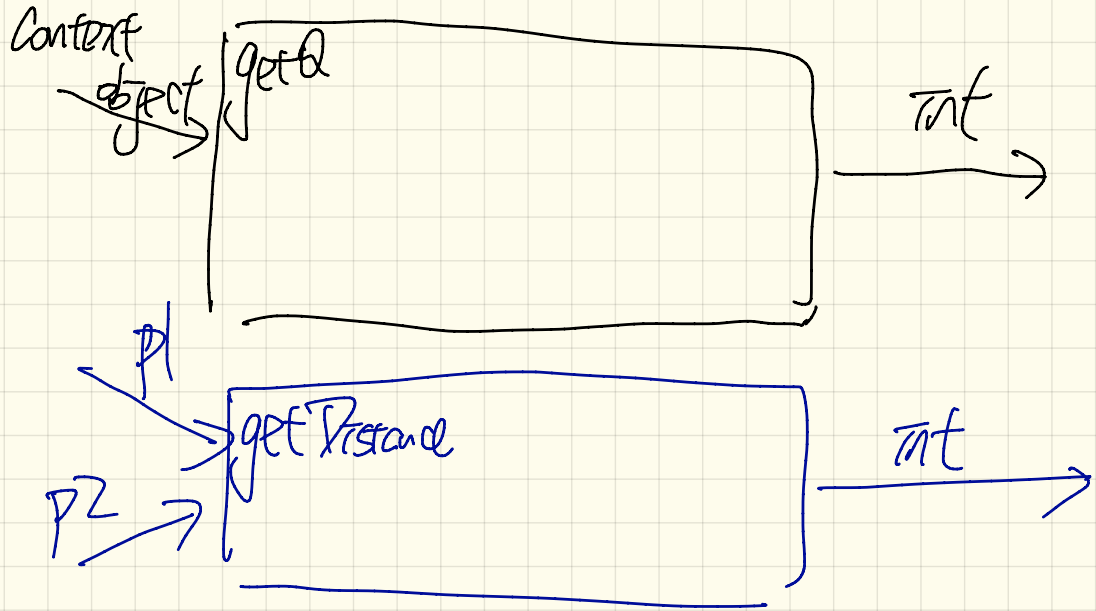
$p2.isAtFQ()$

$p1.getDistance(p2)$

$p1.getDistance(p2, p3)$

$p1.getQ()$	return	1
$0.getQ()$.	0
$p3.getQ()$		4

Method - a block of code
- blackbox



class A {
int l 3

scope
of j

boolean am () {
int j = 3;
l = j * 2; ✓
}

void mm () {
int k = 4;
j l = k * k; ✓
k = j * 2; ✗
}