

Lecture 8

Tuesday Oct. 3

class Point2 {

We declare 'obj' of static type object.

boolean equals (Object obj) {

S.T.

Object p3 =  
new Point2(...)  
D.T.

test() {  
Point2 p1 = new Point2(...);  
Point2 p2 = new Point2(...);  
} static types

dynamic type

Point2 other = (Point2) obj;

return this.x == other.x &&

this.y == other.y;

}

class Point2 {

boolean equals (Object obj) {

return [ this.x == obj.x &&  
this.y == obj.y ];

Object

Point2

double x  
double y

}

}

Fix: Since the previous 3 cases not satisfied, we've  
since 'obj' is a Point2 object x and y. which does not declare

S.T. is Object

```
class Point2 {
```

```
    boolean equals (Object obj) {
```

alternatively:

```
    return  
    this.x == ((Point2) obj).x &&  
    this.y == ((Point2) obj).y ;  
    }  
}
```

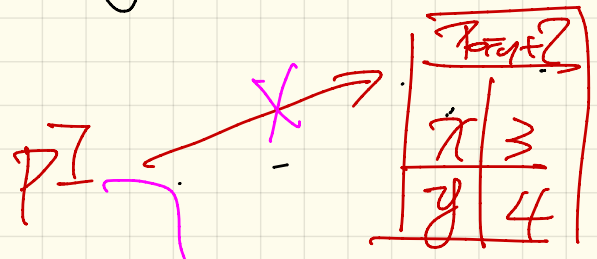
```
        Point2 other =  
            temporarily change  
            the S.T. of obj  
            from Object to  
            Point2  
            (Point2) obj ;  
        this.x == other.x  
        &&  
        this.y == other.y ;
```

once declared, can never be changed

Static type (type during declaration)

Point2 p1

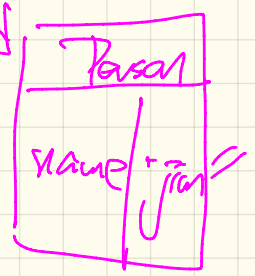
Person jcar



Dynamic type

(1) Point2 p1 = new Point2(3,4);

(2) Object p1 = new Point2(3,4);  
p1 = new Person("jcar");



D.T.

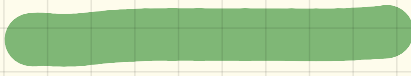
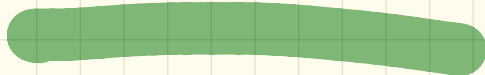
D.T. changes to Person.

Q. Swap the order: `this.getClass() != obj.getClass()`  
|| `obj == null`. ~~X~~ When obj is null → Null Pointer Exception.

```
else if (obj == null || this.getClass() !=  
obj.getClass()) {  
    return false  
}
```

Q. what if obj is null.

A. No. ∴ SCE.

$P$	$Q$	$P \&\& Q$	$P \parallel Q$
F	F		
F	T		
T	F		
T	T		

$P \text{ is } F$ ,

no need to  
evaluate  $Q$

$P \text{ is } T$ ,

no need to  
evaluate  $Q$ .

int  $\bar{i}$  = read from user.

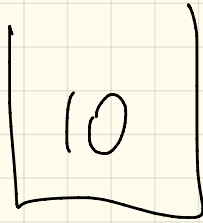
$\bar{i} \in (0 \leq \bar{i} \ \&\& \ \bar{i} < a.length \ \&\& \ a[\bar{i}] > 0)$

Guard to ensure  $\bar{i}$  is a valid index

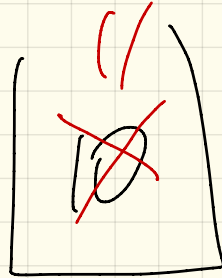
$\bar{i} \in (r \ \&\& \ p \ \&\& \ q)$  X  $\bar{i} < 0$  when  $r < 0$   
 $\bar{i} \geq a.length$  when  $q > a.length$   
r will throw index out of bound



int  $\tau = 10$



$\tau$   
↓  
argument

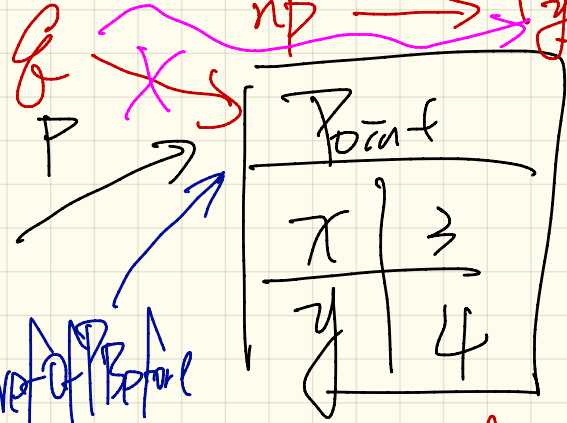


$\tau$   
↓  
parameter

Point P = new Point(3, 4);

Point	
x	6
y	8

Point ref of P Before = P;



U. reassign Ref(P);

assert True (

P == ref of P Before);

wid reassign Ref (Point [q]) {  
 Point np = new Point(6, 8);  
 q = np;

call by ref →