

Monday March 11

Lecture 17

Exam: April 7

Last Class: April 3

Primitive type

char  
boolean  
double  
float  
false true

int

I

= 23

P.getDot()  
P.g.v.

Reference type  
(= address)

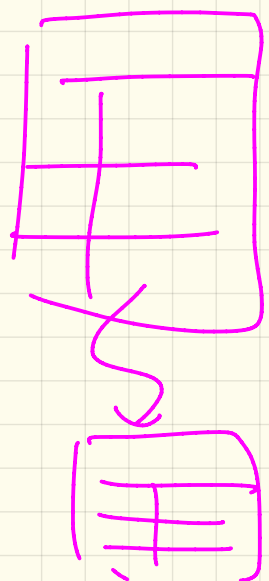
Point

P  
value of address

String  
Scanner

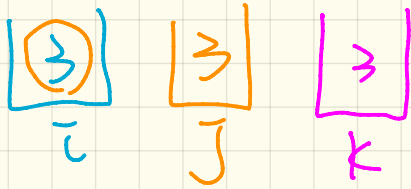
```
class Point {  
    .  
}
```

existing class name



# Copy of Variables : Primitive Type

```
1 int i = 3;
2 int j = i; System.out.println(i == j);
3 int k = 3; System.out.println(k == i && k == j);
```



# Copy of Variable: Reference Type

```

1 Point p1 = new Point(2, 3);
2 Point p2 = p1; System.out.println(p1 == p2);
3 Point p3 = new Point(2, 3);
4 System.out.println(p3 == p1 || p3 == p2);
5 System.out.println(p3.x == p1.x && p3.y == p1.y);
6 System.out.println(p3.x == p2.x && p3.y == p2.y);

```

are p1 and p2 pointing to the same object?

True (T) for p1 == p2

False (F) for p3 == p1 and p3 == p2

True (T) for p3.x == p1.x && p3.y == p1.y

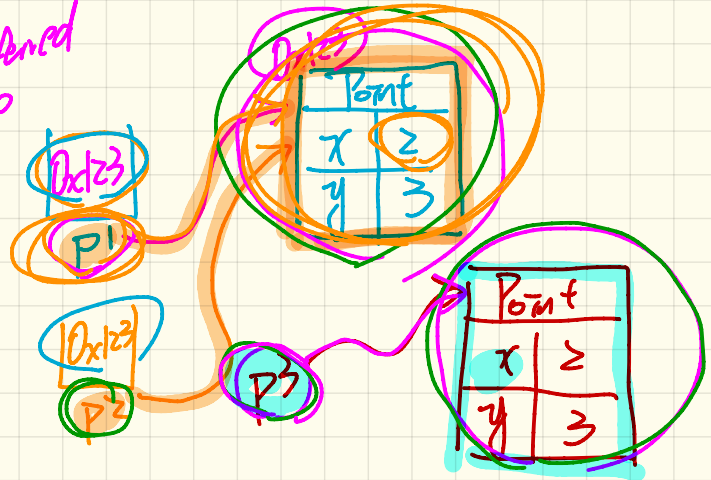
True (T) for p3.x == p2.x && p3.y == p2.y

```

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

```

lookup the object being referred to



Point p1 ↗

Point p2 ↗

⋮

$p1 == p2$

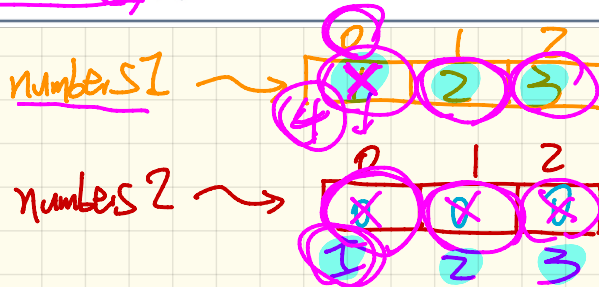
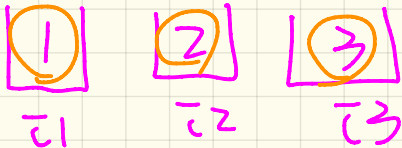
p1 and p2 point to the same object?

$p1.x == p2.x$  &&  $p1.y == p2.y$

**Problem:** Consider assignments to **primitive** variables:

```
1 int i1 = 1;
2 int i2 = 2;
3 int i3 = 3;
4 int[] numbers1 = {i1, i2, i3};
5 int[] numbers2 = new int[numbers1.length];
6 for(int i = 0; i < numbers1.length; i++) {
7     numbers2[i] = numbers1[i];
8 }
9 numbers1[0] = 4;
10 System.out.println(numbers1[0]);
11 System.out.println(numbers2[0]);
```

ns2[0] = ns1[0]  
ns2[2] = ns1[2]



**Problem:** Consider assignments to **reference** variables:

```
1 Person alan = new Person("Alan");
2 Person mark = new Person("Mark");
3 Person tom = new Person("Tom");
4 Person jim = new Person("Jim");
5 Person[] persons1 = {alan, mark, tom};
6 Person[] persons2 = new Person[persons1.length];
7 for(int i = 0; i < persons1.length; i++) {
8     persons2[i] = persons1[i]; }
9 persons1[0].setAge(70);
10 System.out.println(jim.age);
11 System.out.println(alan.age);
12 System.out.println(persons2[0].age);
13 persons1[0] = jim;
14 persons1[0].setAge(75);
15 System.out.println(jim.age);
16 System.out.println(alan.age);
17 System.out.println(persons2[0].age);
```

*array initialized*

0  
persons2[0] = persons1[0]  
1  
2

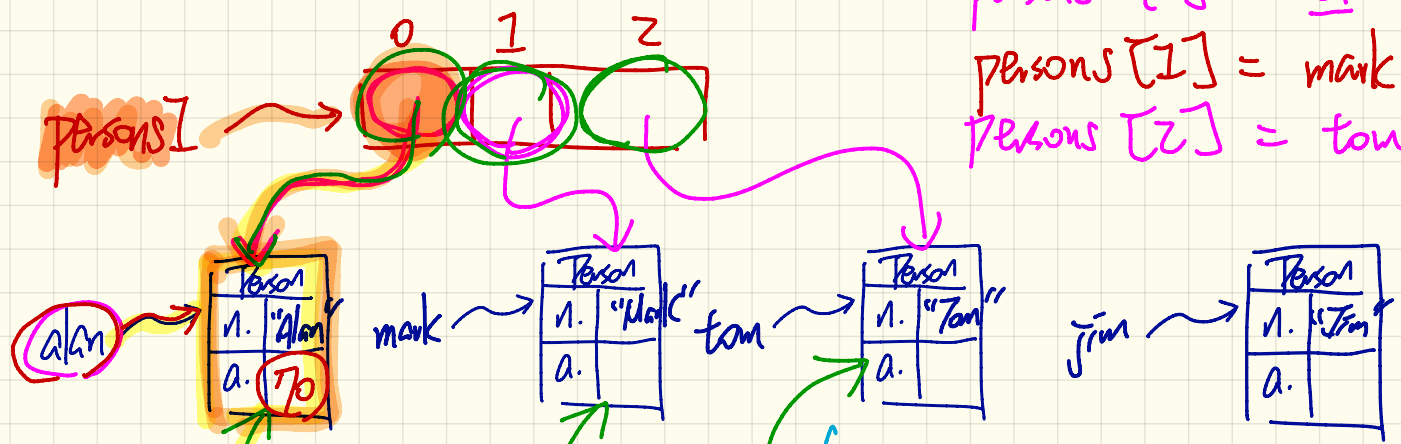


persons[0] = alan ;

persons[0] == alan

persons[1] = mark ;

persons[2] = tom ;



persons2[0] = persons[0]  
persons2[0] == persons[0]



persons[0] == alan

persons[0].getAge() is alan.age 70

array

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

Each element in the array stores the address of some Person object

Person[] persons = new Person[3];

persons[0] = alan; → address

persons[1] = mark;

persons[2] = tom;

int[] iS = { 23, 46, 39 };

- After executing this line: int[] iS = new int[3];
- persons[0] and alan store the address.
  - persons[0] and alan point to the same object.
- iS[0] = 23; iS[2] = 39;
- iS[1] = 46;