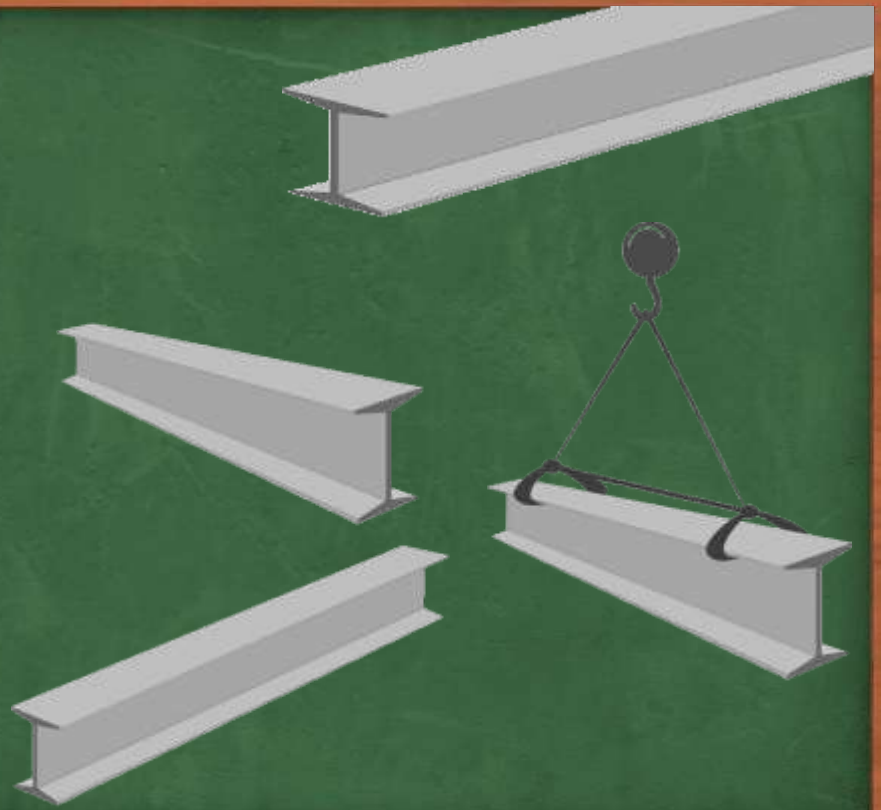
More Structure

- Comparing Objects
- “null” Pointers
- Passing Pointers
- Thinking in Pointers



Introduction to Java

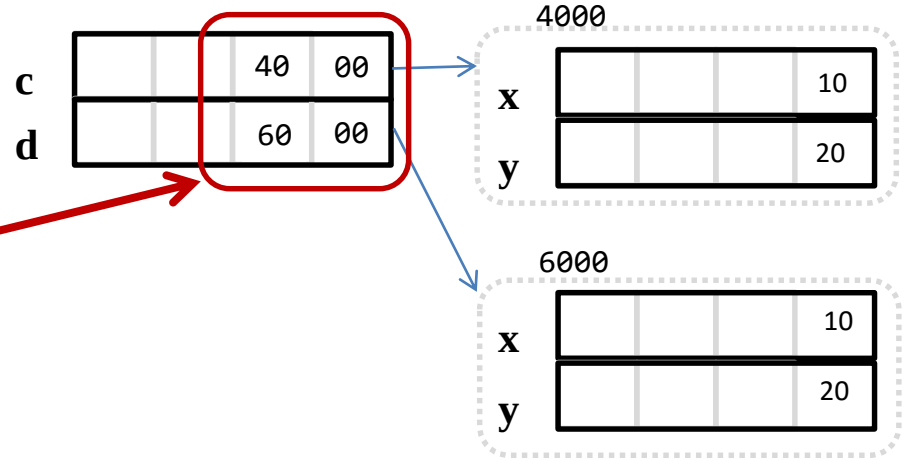
See Also

http://examples.javacodegeeks.com/java-basics/exceptions/java-lang-nullpointerexception-how-to-handle-null-pointer-exception/

https://blog.udemy.com/java-null-pointer-exception/

Compare

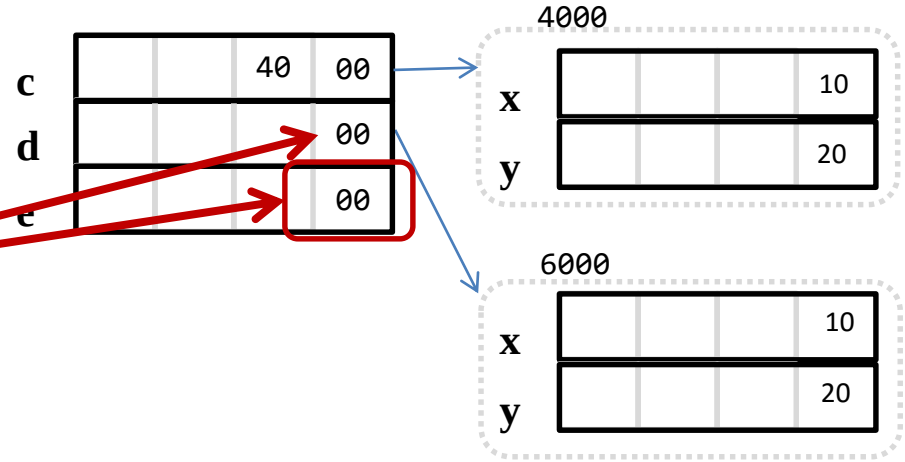
```
Point c = new Point();
c.x = 10;
c.y = 20;

Point d = new Point();
d.x = 10;
c.y = 20;

c = d;

if(c==d) {
    System.out.println("SAME");
}
```



- "=" and "==" operate on the pointer value – not the object
- You have to write code to compare the objects themselves

"null" Pointers

```
Point c = new Point();
c.x = 10;
c.y = 20;

Point d = new Point();
d.x = 10;
c.y = 20;

Point e = null;
d = null;

if(d == null) {
    System.out.println("Nothing");
}
```




```
e.x = 50;
```

OH NO !!!
e is null

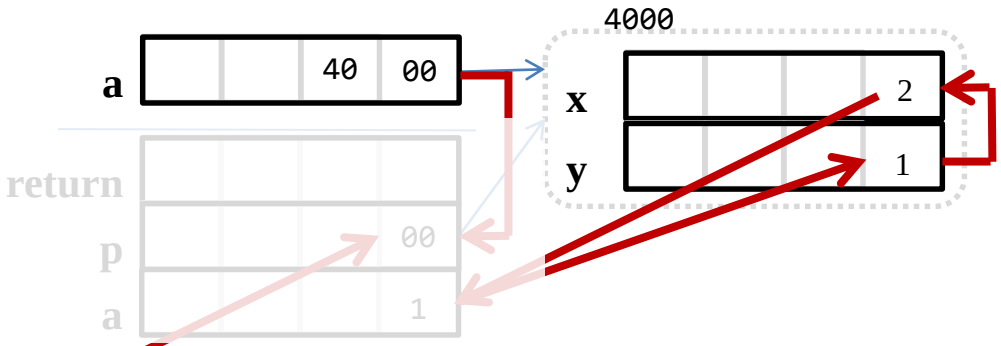
Console

<terminated> Tinker (2) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (Feb 8, 2014, 4:25:14 PM)

```
Exception in thread "main" java.lang.NullPointerException
        at Tinker.main(Tinker.java:26)
```

Passing Pointers

```
public static void main(String [] args) {

    Point a = new Point();
    a.x = 1;
    a.y = 2;

    swapXandY(a);

    System.out.println(a.x); // "2"
    System.out.println(a.y); // "1"

}

public static void swapXandY(Point p) {

    int a = p.x;
    p.x = p.y;
    p.y = a;

    p = null; // take that, caller
}
```



- Pointers are copied like all other primitive types
- Called routine can't change the caller's pointer
- Called routine CAN change the object they share

Object Composition

Point.java

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

Line.java

```
class Line {
    int color;
    boolean visible;
    Point a;
    Point b;
}
```

Triangle.java

```
class Triangle {
    Line t;
    Line u;
    Line v;
}
```

Tinker.java (main)

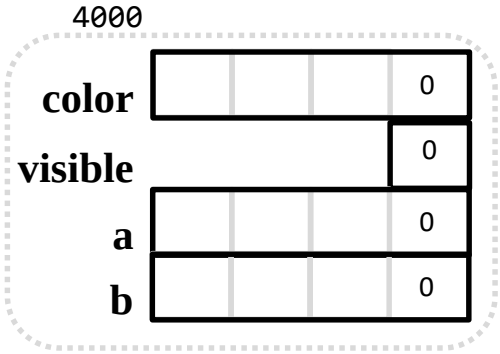
```
Triangle tri = new Triangle();

tri.t.visible = true;
tri.v.a.x = 20;

System.out.println(tri.v.a.x);
```

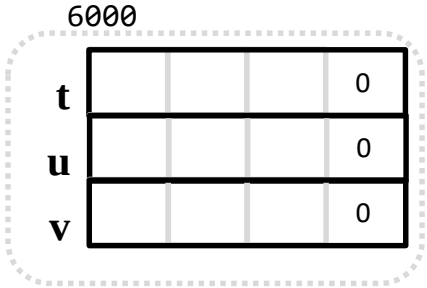
```
Exception in thread "main" java.lang.NullPointerException
    at Tinker.main(Tinker.java:27)
```

- Classes are usually defined in a separate file with the class's name
- Follow the pointers from left to right

Heap Defaults to "0"

```
Line u = new Line();
System.out.println(u.color);   // "0"
System.out.println(u.visible);// "false"
System.out.println(u.a);       // "null"

Triangle tri = new Triangle();
System.out.println(tri.t);    // "null"
```





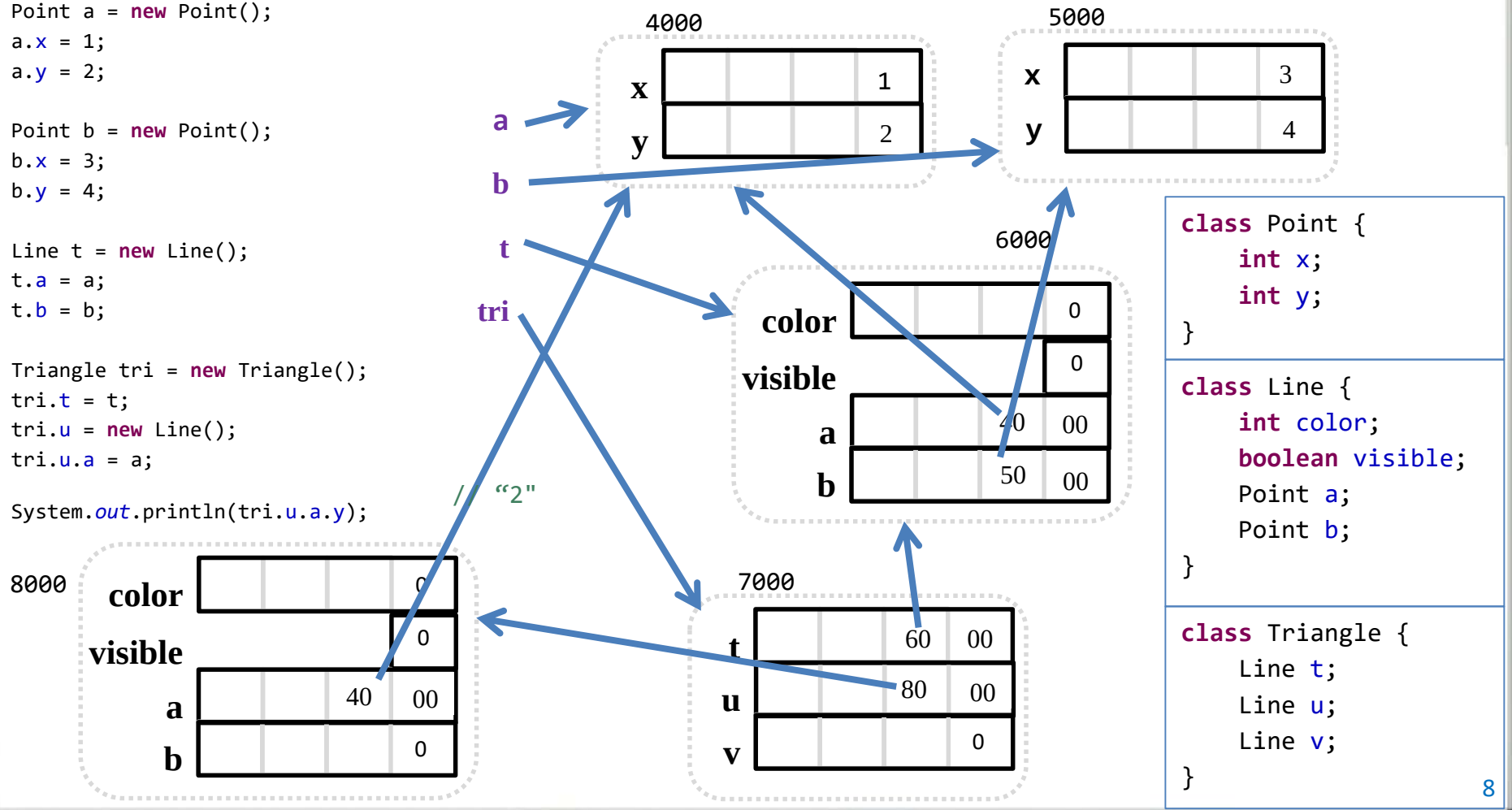

```
class Point {
    int x;
    int y;
}
```

```
class Line {
    int color;
    boolean visible;
    Point a;
    Point b;
}
```

```
class Triangle {
    Line t;
    Line u;
    Line v;
}
```

- "new" heap memory is filled with zeroes
- 0, false, null, etc
- You have to "new" all the pieces manually

```
Point a = new Point();
a.x = 1;
a.y = 2;

Point b = new Point();
b.x = 3;
b.y = 4;

Line t = new Line();
t.a = a;
t.b = b;

Triangle tri = new Triangle();
tri.t = t;
tri.u = new Line();
tri.u.a = a;

System.out.println(tri.u.a.y);
```




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

```
class Line {
    int color;
    boolean visible;
    Point a;
    Point b;
}
```

```
class Triangle {
    Line t;
    Line u;
    Line v;
}
```

Tinkering

- Create the “Triangle.java” from this lesson.
- Create a static function that takes three Points and makes a Triangle.
- Does the Triangle class ensure that the lines touch? Maybe three Lines is not the best way to define a Triangle. Code up a better solution!