EECS1022 Programming for Mobile Computing (Winter 2021)

Q&A - Lectures W8

Monday, March 15

If we have time, can you explain why the default value for a boolean var is false?

Java

boolean bl ;

default value : false

Java → C-family language → EECS 2031 procedural language.

Java

boolean bl = false

bl && . . .

f

when bl's default value is composed with other exp, you may think about if the default false is intended.

std io

bool

b = true ;

b = 0 ; /* false */

b = 3 ; /* true */

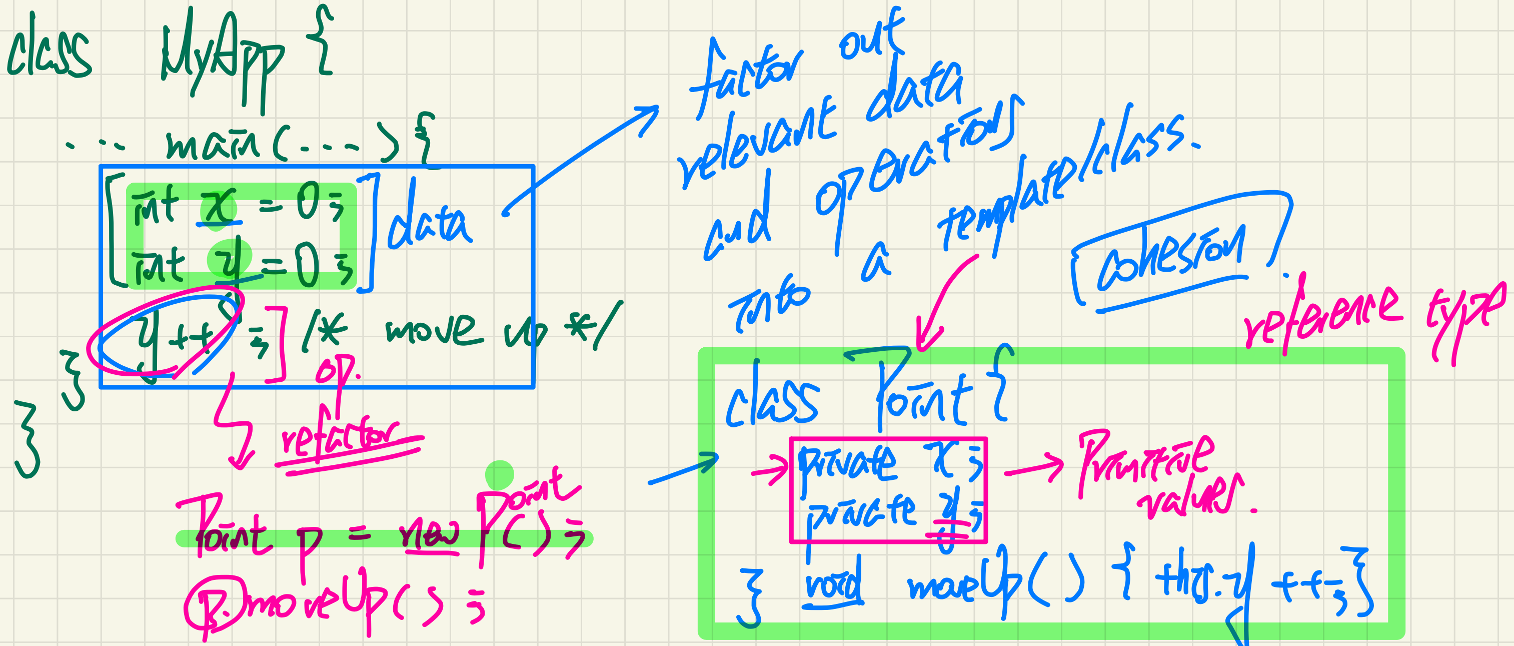
int

0 → false

≠ 0 → true

default value

Is it a correct conceptually to say that reference type data makes use of primitive type data in order for the reference type data to function and serve its purpose? ✓ YES.



```
class MyApp {
  ... main(...) {
    int x = 0;
    int y = 0;   ] data
    y++;  /* move up */   ] op.
  }
}
```

factor out relevant data and operations into a template/class.

cohesion

refactor

```
Point p = new Point();
p.moveUp();
```

reference type

```
class Point {
  private x;
  private y;
  void moveUp() { this.y ++; }
}
```

→ Primitive values.

Reference-Typed Return Values

Is it then right to say that across a class's method types (accessor, mutator), if two or more methods have identical names and parameters despite having different return types, it is invalid?

No.
Methods with the same name must have distinct lists of param. types.

```
public class Point {
  public void moveUpBy(int i) { y = y + i; }
  Point movedUpBy(int i) {
    Point np = new Point(x, y);
    np.moveUp(i);
    return np;
  }
}
```

```
public class PointTester {
  public static void main(String[] args) {
    Point p1 = new Point(2.5, -3.6);
    p1.moveUp(7.8);
    Point p2 = p1.movedUpBy(6.4);
    System.out.println(p1 == p2);
  }
}
```

Reference-Typed Return Values

int getPrice

(a) If we had Point movedUpBy (int y), does this mean the first line of the body has to be something like Point np = new Point(x, this.y)? YES.

(b) Also, just to clarify, the Point np object only has a local scope within the Mutator, correct? YES.

(c) Was the object np created to differentiate p2 from p1 and to ensure p1.x and p1.y are not altered in the invoked accessor method? YES.

```java
public class Point {
  public void moveUpBy(int i) { y = y + i; }
  Point movedUpBy(int i) {
    Point np = new Point(x, y);
    np.moveUp(i);
    return np;
  }
}
```

~~np.~~ - -

→ no modification to 'this' (C.O.)

local var. within the scope of movedUpBy.

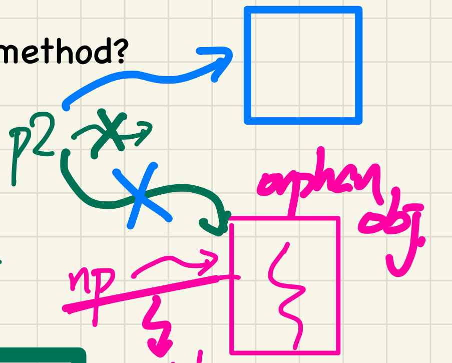
```java
public class PointTester {
  public static void main(String[] args) {
    Point p1 = new Point(2.5, -3.6);
    p1.moveUp(7.8);
    Point p2 = p1.movedUpBy(6.4);
    System.out.println(p1 == p2);
  }
}
```

→ modifying C.O. p1

→ without modifying p1, return the ref. of another obj.

accessor

p2 = new Point(..);



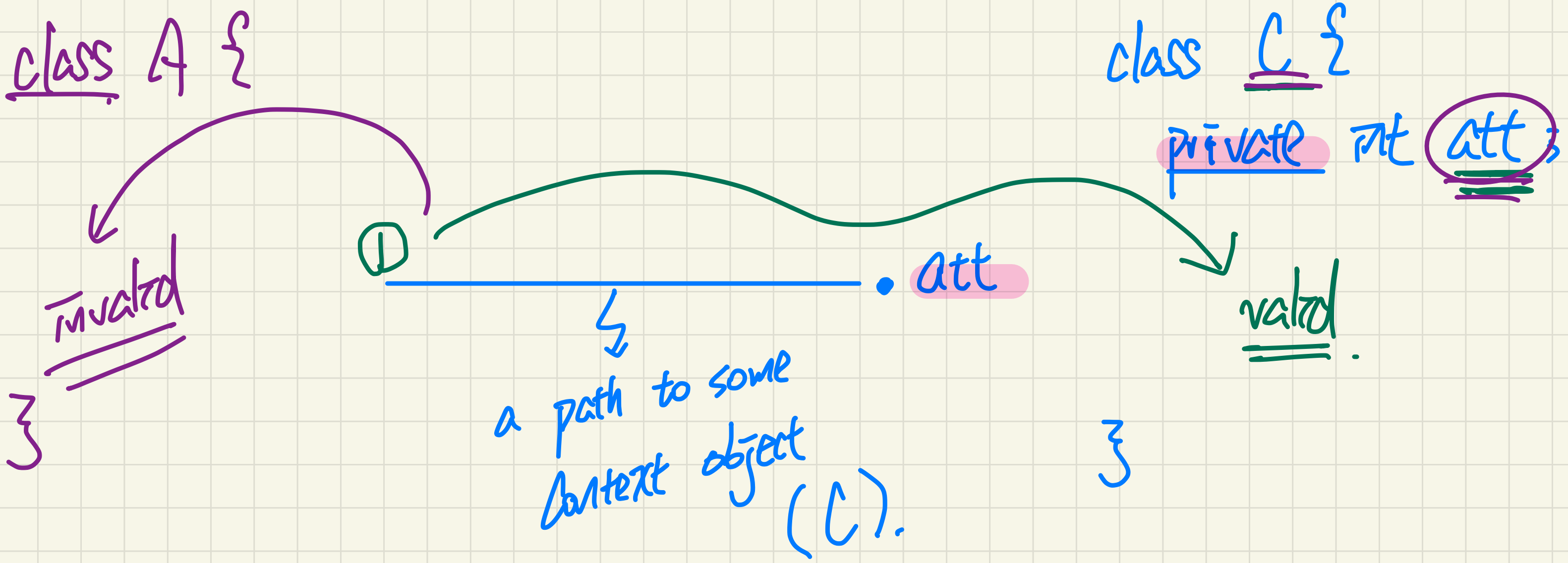
invalid ref ∵ method call ended.

If you create an array with some of the indices storing/pointing to null, will the NullPointerException only occur if and only if you try to invoke a method on those particular indices?

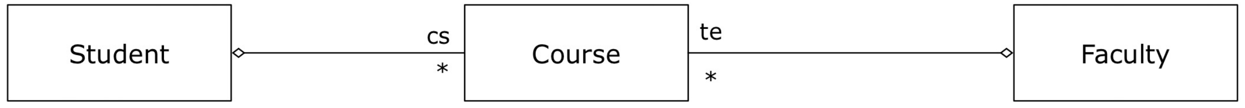
```
Person[] ps = { new Person("alan"), null };

① for( int i = 0 ; i < ps.length ; i++) {
       println(ps[i]);
   }                    null

② for( int i = 0 ; i < ps.length ; i++) {
       println(ps[i].getName());
   }                    null.getName()  NPE
```

Which prog(s) will result in NPE?

If we want to
retrieve private attributes of an object while in the same class,
we can use them directly;
however, if the private attributes are in another class,
we need to use accessor methods i.e. .getName instead of .name?



Dot Notation for Navigating Classes (2)



```
class Student {
  String id;
  Course[] cs;
}
```

✓
```
class Course {
  String title;
  Faculty prof;
}
```

```
class Faculty {
  String name;
  Course[] te;
}
```

```
/* Title of instructor's
 * ith teaching course
 */
String getTitle(int i) {

}
```

choose all expressions Invalid!

Course

not declared

✗ this.cs[i].getTitle();

Faculty

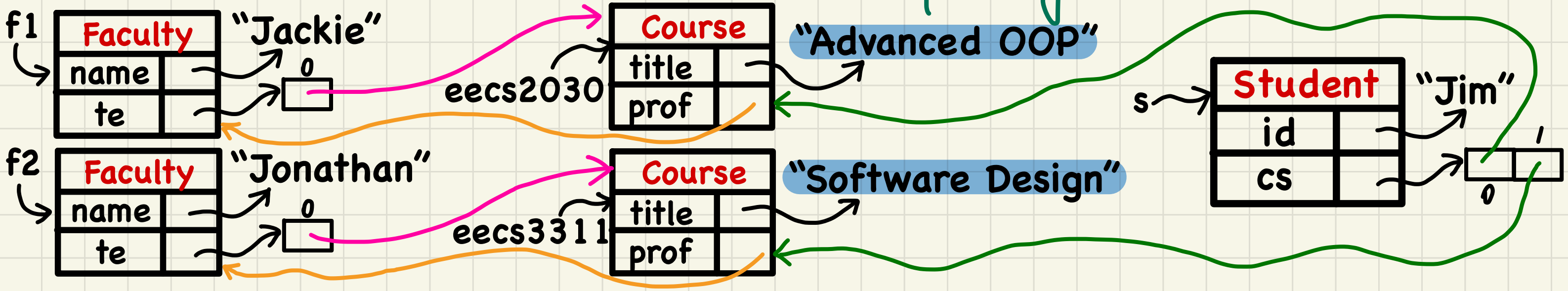
✓ this.prof.getName()

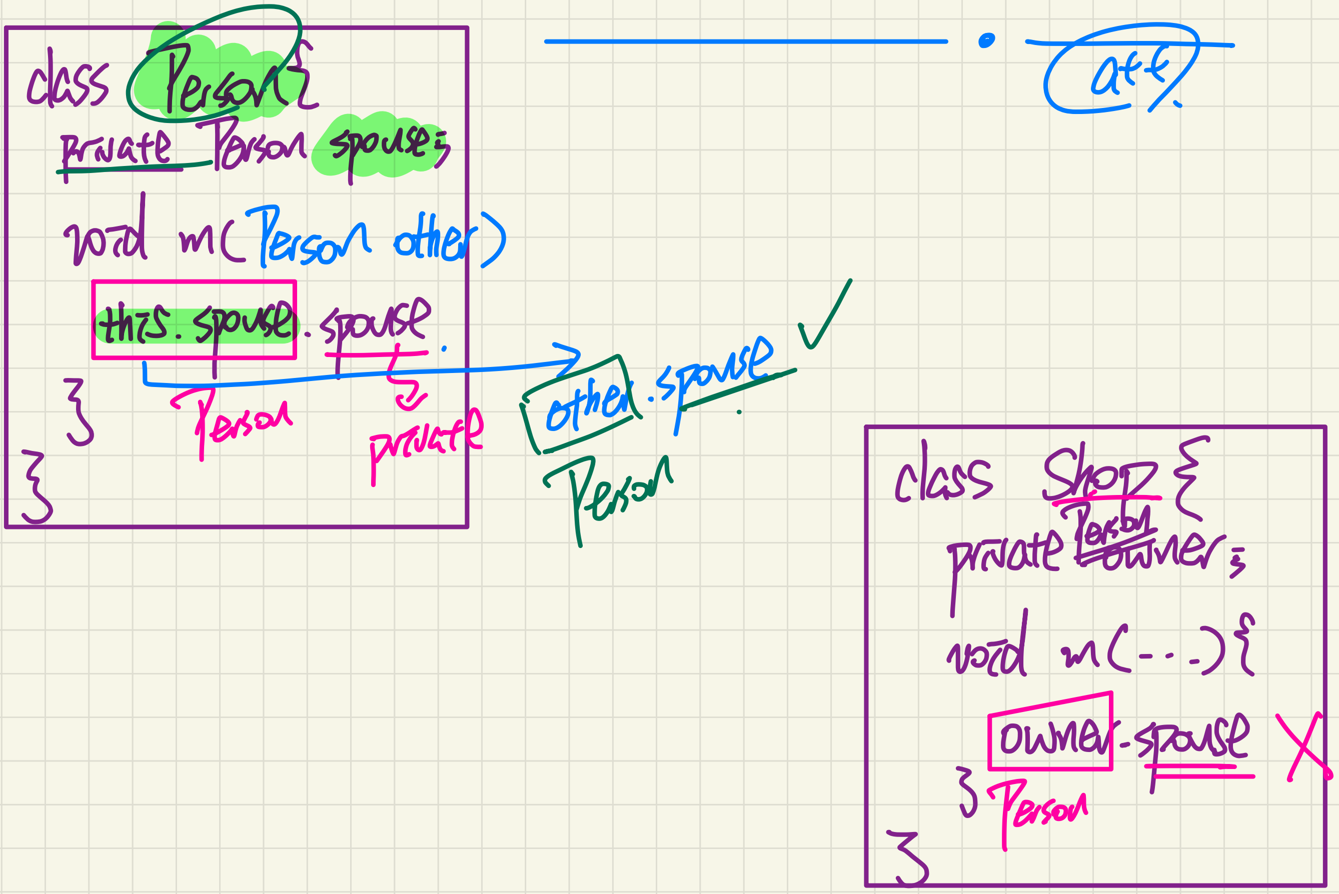
Faculty

✗ this.prof.te[i].getTitle()

te is private and the context class is not Faculty.

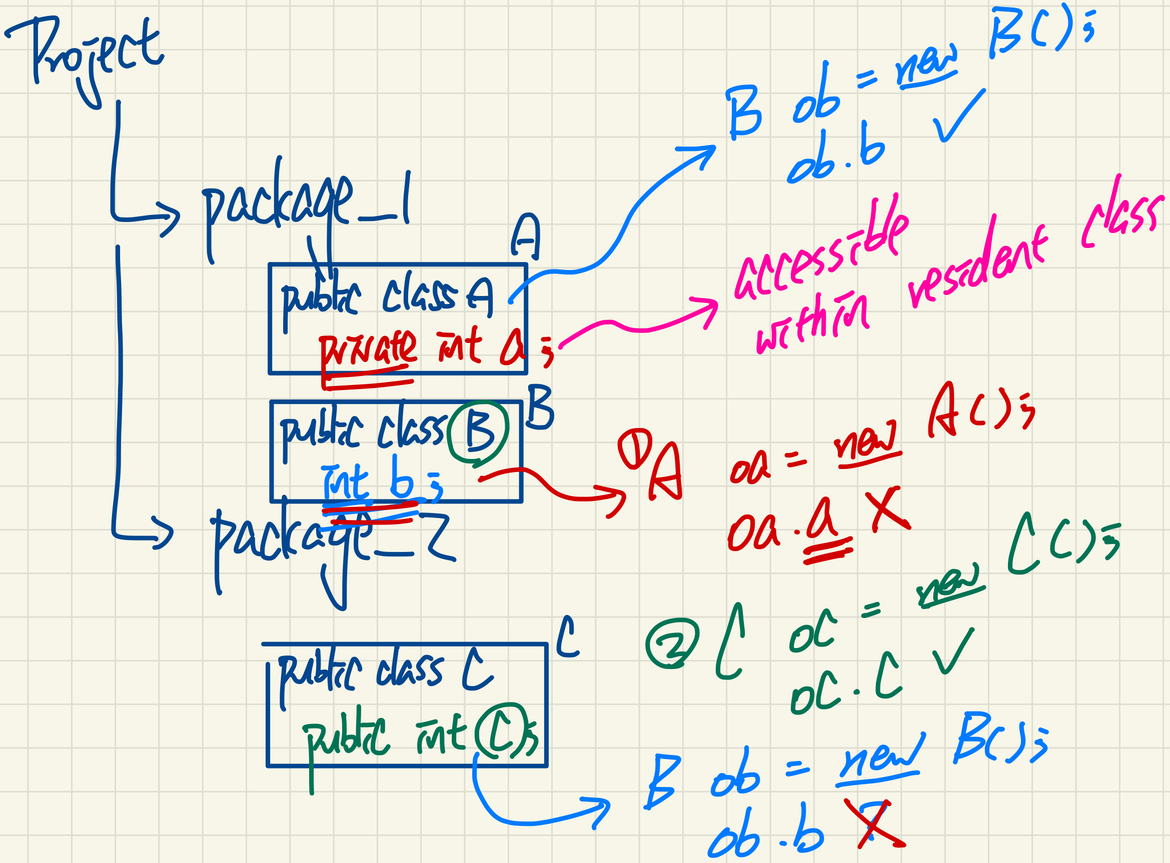
✓ this.prof.getTE()[i].title





```
class Person {
  private Person spouse;

  void m(Person other)
    this.spouse.spouse.
  }
}
```

```
class Shop {
  private Person owner;

  void m(...) {
    owner.spouse   X
  }
}
```



```
B ob = new B();
ob.b ✓

① A oa = new A();
oa.a ✗

② C oc = new C();
oc.c ✓

B ob = new B();
ob.b ✗
```