CSSE 220 Day 5

Objects

Check out *SuperSimpleObjects* and *TeamGradebook* from SVN

Plan for today

- Introduce how to write your own classes
- Talk about object references and box and pointer diagrams
- Get started on TeamGradebook, your new assignment

Identifiers (Names) in Java

- The rules:
  - Start with letter or underscore (_)
  - Followed by letters, numbers, or underscores
- The conventions:
  - `variableNamesLikeThis`
  - `methodNamesLikeThis(…)`
  - `ClassNamesLikeThis`
- You should follow the conventions!

Using Objects and Methods

"Who does what, with what?"

- Works just like Python:
  - *object.method(argument, ...)*

Implicit argument

Explicit arguments

The dot notation is also used for *fields*

- Java Example:

```
String name = "Bob Forapples";
PrintStream printer = System.out;

int nameLen = name.length();
printer.printf("'%s' has %d characters", name, nameLen);
```

Implementing classes

- Live coding with Bank Account object

Constructors

- Called when you create a new instance of an object with `new` e.g.:

```
MyClass var = new MyClass(“hello”);
```

- This implicitly calls a method like this in MyClass

```
public MyClass(String words) {
```

- Use constructors to put your class in a “good state”
- Similar to initializing in Python
- Java implicitly creates a no-argument constructor if you don’t add one

Now code the StudentAssignments class yourself

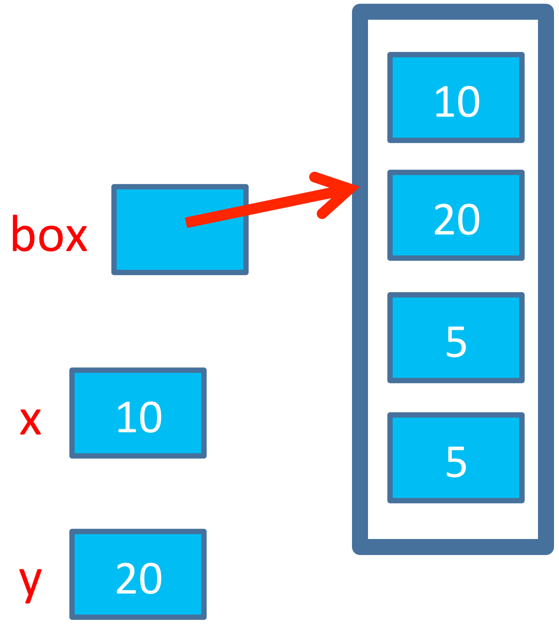
- Uncomment the stuff in StudentAssignmentsMain to see what the class ought to do
- Then create the class and add the constructors and methods you need
- If you finish early, add a function to compute the student's average grade

Differences between primitive types and object types in Java

OBJECT REFERENCES

What Do Variables Really Store?

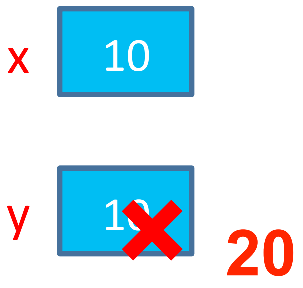
- Variables of primitive type store *values*
- Variables of class type store *references*

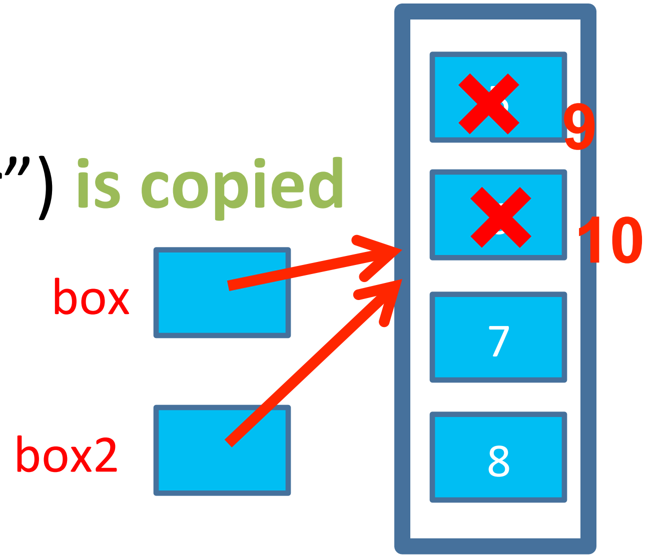


```
1. int x = 10;
2. int y = 20;
3. Rectangle box = new Rectangle(x, y, 5, 5);
```

Assignment Copies Values

- Actual value for number types
- **Reference** value for object types
  - The actual **object is not copied**
  - The **reference value** ("the pointer") **is copied**
- Consider:





```
1. int x = 10;
2. int y = x;
3. y = 20;

4. Rectangle box = new Rectangle(5, 6, 7, 8);
5. Rectangle box2 = box;
6. box2.translate(4, 4);
```

Boxes and lines exercise

Q1 – Q10

Separating implementation details from how an object is used

ENCAPSULATION

Encapsulation in Object-Oriented Software

- *Encapsulation*—separating implementation details from how an object is used
  - Client code sees a *black box* with a known *interface*

	Functions	**Objects**
Black box exposes	Function signature	Constructor and method signatures
Encapsulated inside the box	Operation implementation	**Data storage** and **operation implementation**

Start on TeamGradebook

- Try to finish the code for both add-student and get-names today
- If you are confused about what to do, get help!

Q11