On your way in...

Pick-up
1. Nothing!
2. (Reuse Lecture #29 Notes, see the back)
A bunch of rough drafts of new POGIL activities now up on the course website. Let me know if there are issues.

Tuples, Dictionaries, Generators, List Comprehensions, LinkedList Elements & Wrappers.
JAVA

Classes in Java class.
THIS WEEK’S JAVA LAB.

Pro Tip: Read this week’s lab document carefully. Lots of Java insight.
Java - Classes

• Building the Color Class in Java

• See example code:
  - cd ~/cs134/shared/examples/04.29
  - git pull
  - emacs Color.java
Java – Things to Notice

• Private instance variables instead of __slots__
• Constructor with no return type, __init__
• this instead of self
  ▪ Don’t need to pass this as an argument. Implicit!
• No default values for parameters
  ▪ Can have multiple functions with same name, but diff # parameters
• No properties, but accessor methods instead
Java – Things to Notice

• boolean instead of bool
  ▪ true, false instead of True, False

• && instead of and

• || instead of or

• ! Instead of not

• Python:  a is not b
• Java:      a != b
Java – Things to Notice

- null instead of None
- == instead of is
  - Comparing strings: string1.equals(string2)

- public String toString() → Sort of like __str__(..)

- new when instantiating a new object:
  - Tree alice = new Tree(“alice”)
Java – Things to Notice

• `javap java.lang.Math`
  ▪ Instead of pydoc3

• We use the Javadoc API
  ▪ [https://docs.oracle.com/javase/8/docs/api/index.html](https://docs.oracle.com/javase/8/docs/api/index.html)

• Like python, can make comments that generate javadoc for your code
Java – static

• Shared across all instances
  ▪ Vs. dynamic: operates on a given instance

• Considerations for when to use static methods:
  ▪ Doing something that doesn’t change across instances
  ▪ OR Not using any instance variables
  ▪ OR not dependent on instance creation
    ▪ Does it makes sense to call this method even if no object has been constructed yet?

We’ll use static methods for our q20.java program in lab
Java – Things to Notice

- Indenting’s not important for semantics, but it is for **legibility**
  - In emacs: highlight region to format
  - `M-x indent-region` will format the highlighted area appropriately

- Use single-line comments to comment complex regions of code
  - `// Here is a one-line comment`

- Use multi-line comments to describe function behavior
  - `/* Multi-
    Line
    Comment
    */`
Example Code

- cd ~/cs134/shared/examples/04.29
- git pull
- emacs Color.java
Java: Language Basics

• Java Docs Tutorial:
  • [https://docs.oracle.com/javase/tutorial/java/nutsandbolts/index.html](https://docs.oracle.com/javase/tutorial/java/nutsandbolts/index.html)

• Java Docs API (like pydoc3)
  • [https://docs.oracle.com/javase/8/docs/api/index.html?overview-summary.html](https://docs.oracle.com/javase/8/docs/api/index.html?overview-summary.html)
QUESTIONS?
Leftover Slides
Our First Java Program

public class first {
    public static void main(String[] args) {
        // Comment! Do something
    }
}
Running a Java Program vs. Python

1. Compile the java program into machine code
   - `javac first.java`
   - This produces `first.class`

2. Run the machine code
   - `java first`
   - Java Virtual Machine runs the code

1. Call python
   - `python3 first.py`
   - Python turns code into bytecode and interpreter interprets it to the machine
Java – Primitive Types

- Everything is **not** an object in Java! Unlike python
- Primitive types don’t have methods
- Primitive types
  - `int`, `boolean`, `double`, `char`, `float`, etc.
Double vs. Float

- **double** occupies 8 bytes versus 4 bytes for a **float**
- Floating point arithmetic with **double** values is slower than with **float** values
  - (hardware dependent)
- **double** can represent larger (and smaller) numbers than a **float**
- **double** represents numbers with more than twice the precision of a **float**

The simple "rule of thumb" is to use **double**, not **float**, UNLESS there is an **overriding** performance requirement, AND you have solid evidence that using **float** will make a difference with respect to that requirement.