On your way in...

Pick-up
1. Lecture 32 Notes

Lab 8 Supreme Court Analysis grades are available
1. cd ~/cs134/lab8
2. git pull
3. emacs GradeSheet.txt
HW11 DUE MONDAY.

+ If you finished your Java Lab 10, then you don’t need to come to lab.

After Monday/Tuesday, we’re done all assignments.
Welcome to CS 134!

Introduction to Computer Science
Iris Howley

- JAVA IV -

Spring 2019
JAVA

Subtypes and Supertypes.
Subtypes

Object

Automobile

Mazda  public  void  drive()

Tesla  public  void  charge()
Tesla is a *subtype* of Automobile. Automobile is Mazda’s *supertype*.

Object is the top-most *supertype* of all objects.
Subtypes

Object

Automobile

Mazda

Tesla

public void drive()

public void charge()

Mazda mine = new Mazda()

**mine doesn’t have a method .charge() !**
Subtypes

Object

Automobile

Mazda

Tesla

public void drive();

public void charge();

Automobile a = new Tesla();

It’s always legal to assign a subtype value to a supertype variable. (the reverse is dangerous)
JAVA

Building our own List data structure.
Java - Arrays

- `int[] data = new int[100];`
- `System.out.println(data.length + " vs 100.");`
- `System.out.println(data[50]);`

- 100 vs 100.
- 0

- Maybe we want something more like a Python list, where we create a new list and it’s empty!
- Still have to allocate memory for the list
Java - Arrays

This is the memory ALLOCATION

- `int[] data = new int[100];`
- `System.out.println(data.length + " vs 100.");`
- `System.out.println(data[50]);`

- 100 vs 100.
- 0

We also want a list SIZE for how many elements are actually in the list

We don’t want to be able to access an item at index 50, if we haven’t added an item there!
List.java

- cd ~/cs134/shared/examples/05.03
- emacs List.java
QUESTIONS?
Leftover Slides
Programming with Assertions

• See documentation:
  ▪ [https://docs.oracle.com/javase/7/docs/technotes/guides/language/assert.html](https://docs.oracle.com/javase/7/docs/technotes/guides/language/assert.html)
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
  - [https://www.oracle.com/technetwork/java/javase/documentation/index-137868.html](https://www.oracle.com/technetwork/java/javase/documentation/index-137868.html)
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)
Java Arrays

• Like python lists, but not

• See example code:
  ▪ cd ~/cs134/shared/examples/05.01
  ▪ git pull
  ▪ emacs StringArrays.java
Java – Arrays

• See documentation:
  - https://docs.oracle.com/javase/tutorial/java/nutsandbolts/arrays.html

• Declared in 2 ways:
  1. `double[] data = new double[100];`
     - One hundred 0.0s
  2. `double[] data = {1.0, 3.0, 5.1};`
     - Array contains 3 decimal/double values
Java – Arrays

• Iterating over arrays:
  - double[] data = {1.0, 3.0, 5.1};

  1. for (int i=0; i<data.length; i++) {
      System.out.println(data[i]);
  }

  2. for (int value : data) {
      System.out.println(value);
  }

Documentation: https://docs.oracle.com/javase/tutorial/java/nutsandbolts/arrays.html
Java – Arrays

• You cannot append values beyond the initial declaration size

• `double[] data = new double[10];`
  ▪ data can contain at most 10 items
  ▪ If you need more, you need to copy the values into a new array
  ▪ This can get expensive (time-wise)!

Documentation: https://docs.oracle.com/javase/tutorial/java/nutsandbolts/arrays.html