On your way in...

Pick-up
1. Lecture 30 Notes
2. HW 11 (due Monday)
3. Graded HW 9 (front desks)

Drop-off
1. HW 10 (2 piles, < 50)
POGIL.

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
Welcome to CS 134!

Introduction to Computer Science
Iris Howley

- JAVA III -
JAVA

Strings and Arrays.
Java Strings

• Similar, but different from python

• See example code:
  - cd ~/cs134/shared/examples/05.01
  - git pull
  - emacs StringArrays.java

• See example code:
  - cd ~/cs134/shared/examples/05.01
  - git pull
  - emacs WordFreq.java
Java – Strings

• See documentation:
  - https://docs.oracle.com/javase/8/docs/api/java/lang/String.html
• Immutable, just like in python

• Concatenation: `myString = “a” + “bc” + 3` → “abc3”
  - `myString.length()` → 4
  - `py: len(myString)`
• `myString.charAt(1)` → ‘b’
  - `py: myString[1]`
WHEN COMPARING STRINGS, WE USE `.equals()` TO CHECK IF THE STRINGS HOLD THE SAME VALUE.

== IS ONLY USED FOR THE PYTHON EQUIVALENT, is

```
s1 = "hello"
s2 = "hello"
s3 = "goodbye"

s1.equals(s2) → true
s1 == s2 → false
```

mysting1.equals(mysting2)
Java – Strings

- `myString = "abc3"

- Substring: `sub = myString.substring(1, 3)`
  - `sub \rightarrow "bc"
  - No string slicing! No negative indices!

Documentation: https://docs.oracle.com/javase/8/docs/api/java/lang/String.html
Java – Strings

• `myString = “abc3”`

• Locate/Find: `foundIndex = myString.indexOf(“bc”)`
  ▪ `foundIndex -> 1`
  ▪ `foundIndex = myString.indexOf(“wut”) -> -1`

Documentation: https://docs.oracle.com/javase/8/docs/api/java/lang/String.html
Java – Strings

• `myString = "a-b-c-3"

• Split: `s = myString.split("-")`
  ▪ `s → ["a", "b", "c", "3"]`

• Trim (py: `strip()`): `s = myString.trim()`
  ▪ `s → "a-b-c-3"

• Join: `s = String.join("+", "will", "I", "ams")`
  ▪ `s → "will+I+ams"`

Documentation: https://docs.oracle.com/javase/8/docs/api/java/lang/String.html
Java – Strings

• myString = “a-b-c-3”

• Starts With: s = myString.startsWith(“a-”)
  ▪ s → true
  ▪ s = myString.startsWith(“wut”) → false

• Ends With: s = myString.endsWith(“-3”)
  ▪ s → true
  ▪ s = myString.endsWith(“wut”) → false

Documentation: https://docs.oracle.com/javase/8/docs/api/java/lang/String.html
Java – Strings

• `myString = “aBcD3”`

• Lowercase: `s = myString.toLowerCase()`
  ▪ `s \rightarrow “abcd3”`

• Uppercase: `s = myString.toUpperCase()`
  ▪ `s \rightarrow “ABCD3”`

Documentation: [https://docs.oracle.com/javase/8/docs/api/java/lang/String.html](https://docs.oracle.com/javase/8/docs/api/java/lang/String.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];`
     o One hundred 0.0s
  2. `double[] data = {1.0, 3.0, 5.1};`
     o 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
Java Vectors

- See documentation:
  - https://docs.oracle.com/javase/7/docs/api/java/util/Vector.html

- Ordered list, like an array, but extensible
- Can get the size of a vector with .size() method

- See example code:
  - cd ~/cs134/shared/examples/05.01
  - git pull
  - emacs WordFreq.java
Java – Vectors

- Vector<String> v = new Vector<String>("hello", "hi");
- sz = v.size(); // Will be 2

- Append values
  - v.add("hey");
  - v.add(1, "heya"); // inserts and shifts

Documentation: https://docs.oracle.com/javase/8/docs/api/index.html?java/util/Vector.html
Java – Vectors

• Vector<String> v = new Vector<String>("hello", "hi");
• v.add("hey");

• No square bracket indexing!
  ▪ first = v.elementAt(0);
  ▪ second = v.get(1);

• Changing values
  ▪ v.setElementAt("yo", 2);
  ▪ v.set(1, "whatsup"); // Will also return original value
Java – Vectors

• Find values
  ▪ `loc = v.indexOf("hello");` // will be 1
  ▪ `loc = v.indexOf("goodbye");` // will be -1 if not found
  ▪ `exists = v.contains("yo");` // boolean found/not

Documentation: https://docs.oracle.com/javase/8/docs/api/index.html?java/util/Vector.html

Lots of other operations!
See the Java documentation!
Java Strings, Vectors

• See example code:
  - `cd ~/cs134/shared/examples/05.01`
  - `git pull`
  - `emacs WordFreq.java`

• There’s also a python version of the code:
  - `emacs WordFreq.py`
  - First approach is a replication of Java code
  - Second approach is more pythonic, better python programming
WordFreq.java & WordFreq.py

• Have a list of objects that store word→count
• Outer loop: look at each word in the file
  ▪ Inner loop: check our wordCounts to see if we’ve already counted it
    ○ (add one if we have!)
  ▪ Add new word→count of 1, if we haven’t counted it yet

• Print each word and its number of occurrences
QUESTIONS?
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: 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)