CS212

Packages

http://java.sun.com/docs/books/tutorial/java/interpack/packages.html

Motivation

- Programming as language
  - alphabet, words, sentences, paragraphs, ...
  - what's next?
- Some software principles:
  - abstraction
  - modularization
- Examples:
  - functions, classes
  - MATLAB tool box
  - "include" files

The Gist

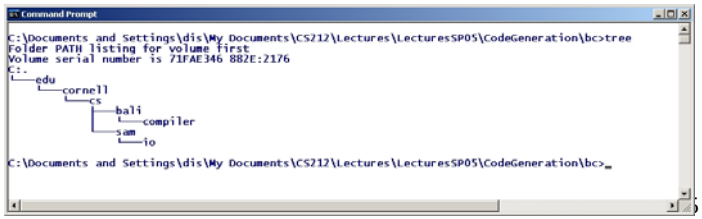
- Java?
  - ***packages***: http://java.sun.com/docs/books/tutorial/java/interpack/packages.html
  - "A package is a **collection of related classes and interfaces** providing access protection and ***namespace*** management."
  - What's namespace? (coming up)
- Examples:
  - Java API:
  - So, what does it mean to import `java.util.*`?

Names

- Namespace:
  - set of names in which each name has a unique meaning
  - related to context...context determines meaning
  - example:
    - in the context of the Internet, a link is a connection to another URL
    - in the context of a chain, a link is one ring that connects to at least one other link
- Some related concepts:
  - ***Visibility***: can a name be accessed?
  - ***Scope***: location in code where name is visible

Package Naming

- **another view of a package**: directory that contains bytecode of classes and interfaces
- naming:
  - ***dir1.dir2.***..
    - example) **java.lang.Math**
    - Package **java** contains package **lang** contains class **Math**
- each dir must be an *actual* dir:
  - example) see package **edu.cornell.cs.sam.io**:



Putting Things in Packages

- All classes and interfaces go in a package
  - what about all work you've done so far?
  - if you have **CLASSPATH** set to **.**, the current directory is your default package
- Use package statement(s) at top of Java class/interface:
  - **package *packagename;***
  - ***packagename*** includes the directory path

6

Accessing Package Members

- Fullname for accessing a particular class:
  - before class definitions: **import *package;***
    example) **import java.lang.Math;**
  - can actually use fullname through code to avoid import statement
- Can import all package members:
  example) **import java.util.*;**

7

Visibility Modifiers

- **protected**:
  - member is visible by all classes in same package and all subclasses of member's class
  - subclass can be in different packages
- default visibility ("blank" modifier):
  - we "fake" **public** access with it
  - actually more restrictive than **protected**!
  - cannot access member from outside package

8