
Held: Tuesday, March 9, 2010

Summary: We consider a variety of recent issues, include dynamically-loaded code, makefiles, and file comparisons.

Related Pages:
- EBoard.

Notes:
- I’ve made a few updates to the schedule. I’m also making some updates to the grading policies, since the course has taken a bit of a different tack than I had planned.
- Reading for Thursday: Thinking in Tkinter Send questions.
- No BC reading for Thursday.
- Thursday will be a “headless lab”. I’d recommend that you all do it during class time, but you may find it useful to do it at other times.
- Assignment for next Tuesday: A preliminary UML architecture for the project.
- We may want to look at The source code for DiffMerge

Overview:
- Quick project discussion.
- Dynamically loading modules in Python.
- Dynamically loading modules in Java.
- Fun with Make.
- Beautiful Code: Code in motion.

Quick project discussion
- We will regularly have three teams of four students, but the particular team membership may vary from subproject to subproject.
- The first teams will be responsible for the preliminary UML designs due next week.
- How do you want to form the first set of teams?

Dynamically loading code
- Friday’s CS Table: Why systems that dynamically load modules are better.
- Our task: Write a small system that dynamically loads modules.
- In particular: We will repeatedly prompt the user for a class name, build an object in that class (if possible), and call the object’s run method.
• We will try to do so in Python and Java. (Dynamic loading can be a PITN in C.)
• Goal: Under an hour.
• Step zero: Make sure that we understand the purpose (simple use cases).
• Step one: Design unit tests.
• Step two: Outline our process in pseudocode.
• Step three: Turn the primary parts of the process (other than the actual module loading) into code.
• Step four: Figure out how to load a module.
• Step five: Hands-on testing.
• Step six: Unit tests.

**Dynamically loading modules in Python**

• How might you approach this problem? What techniques do you know? Where would you look for ideas?

**Dynamically loading modules in Java**

• How might you approach this problem? What techniques do you know? Where would you look for ideas?

**Fun with Make**

• We’ll go over the assignment for today, considering some of the problems you had (and some of the problems that you didn’t know that you had).

**Beautiful Code: Code in motion**

• Primary thesis?
• What is beautiful?
• What is ugly?