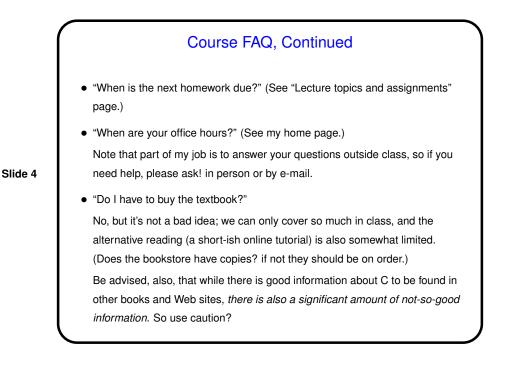


Slide 3

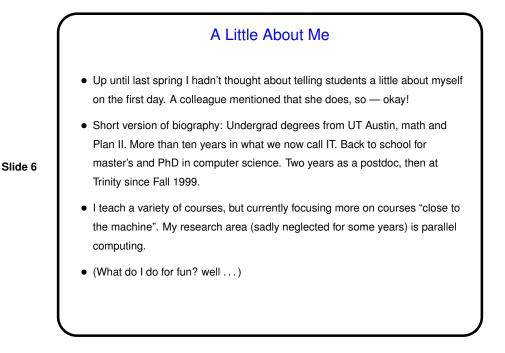
syllabus.)



Course FAQ, Continued

• "What computer(s) can I use to do homework?"

Easiest option may be department's Linux machines. You should have physical access via your TigerCard to all the classrooms and labs (probably not today but soon) any time the building is open. You should also be able to log in remotely to any that are booted into Linux, or to a cluster of Linux-only machines in ITS's server room (names diasnn, where nn ranges from 01 to 05). To log in from off-campus, we are currently recommending that you use ITS's VDI.

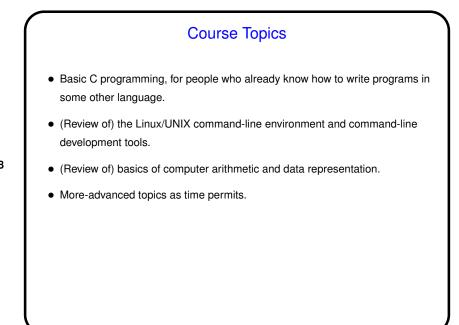


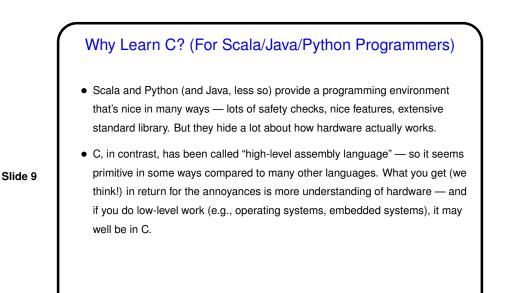
What Is This Course About?

• Back story: Primary goal of our traditional first course (CSCI 1320) is to introduce students to programming and algorithmic problem-solving. Another goal of the course as taught some years ago, however, was to expose students to certain low-level concepts that contribute to a well-rounded education in computer science. Students coming into the major via other routes often did not get this exposure and struggled in later courses.

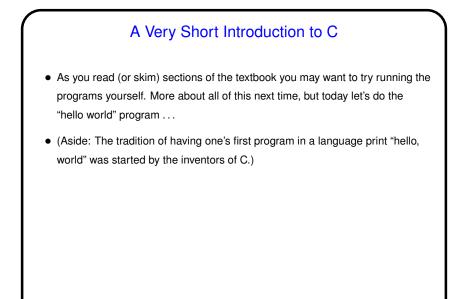
Slide 7

• CSCI 1120 was added to the curriculum as a way to address this problem i.e. to cover the parts of CSCI 1320 that might not be covered by alternative introductory courses. A few years ago we switched to a more-abstract language for CSCI 1320, and at that point this course became required for all students.





First Things First(?) — Text Editors
In class I will use vim to write programs. I don't insist that you use it too, but it's a good tool for this job, and if you aren't very good with it, no time like the present to get better with it. To encourage you, see the first homework.
Slide 10
(Indeed, I think this class is a good time to get more practice with the command line in general; it's in keeping with the spirit of the course, and you have an instructor who knows it pretty well.)



Slide 11

A Very Short Introduction to C, Continued First write the program using a text editor (e.g., vim) and save it with a name ending in .c (say hello.c). (See the "sample programs" Web page for what it looks like.) Next, compile the program (turn it into something the computer can execute). Simplest command for that: gcc hello.c If there are no syntax or other errors, this produces an "executable" file a.out. Run the program by typing ./a.out at the command prompt.

Minute Essay

 (Most lectures will end with a "minute essay" — as a quick check on your understanding, a way for me to get some information, etc., and also to track attendance. Just put your answer in the body of the message; no Word documents please, and put "minute essay" and the course in the Subject line.)

- Tell me about your background: If you took CSCI 1320 at Trinity, who was your instructor? Do you have other programming experience?
- What are your goals for this course? Anything else you want to tell me? (Maybe something interesting you did over the break?)
- Don't forget the reading and homework for next time