Slide 1

Administrivia

- (None?)

Slide 2

Minute Essay From Last Lecture

- About that quiz question, some people did run out of time; others reported other problems. “Hm!”?
- One person asked about whether `struct`s were used in Python. My answer — if it’s implemented in C (possible), yes in that sense, but in any case many programming languages have something similar, and “object-oriented” languages have something more featureful.
- (“Implemented in C”? yes . . .)

Slide 3

Digression — Implementing Programming Languages

- For all programming languages there has to be *something* that translates source code into machine language (remember that from earlier?) — “compiler” for many, “interpreter” for others.
- Such tools are often written in C, with library functions written in, well, it varies — many will be in the language itself (e.g., C for C, Java for Java, etc.) and others in assembly language (semi-human-readable form of machine language).
- But then C itself . . . ? “Bootstrap” problem, usually solved by first writing a very simple version of the compiler in assembly language, and then using it to compile more-complex version. “Hm!”?

Slide 4

`struct`s in C — More Examples

- One example: Convert month, day, year into Julian date (year and day in year). Seems to make sense to use a `struct` to represent month with name and days in month. (Show example code.)
- Another (somewhat contrived) example: Given an input file of integers and a set of integer “ranges”, say how many elements are in each range. Might make sense to use a `struct` to represent range as starting and ending values. Could even sort them with `qsort`(!). (Show example code.)

Slide 5

Minute Essay

- Can you think of uses for `struct`s?