

Slide 1

Administrivia

- Reminder: Homework 5 on Web; due next Monday.
- (Discuss syllabus revisions/corrections.)

Slide 2

Function Pointers

- You know from Java that there are situations in which it's useful to have method parameters that are essentially code (e.g., GUI listener methods, `compareTo` method for sorting, `run` method for threads).
- In Java, you often do this by way of a class whose main or only purpose is to hold the needed code.
- In C, however, you can explicitly pass a pointer to the function.

Function Pointers in C

- The type of a function pointer includes information about the number and types of parameters, plus the return type.
- Example — last parameter to library function `qsort` (in its man page). Call this by providing, in your code, a function with declaration

Slide 3

```
int my_compare(const char *, const char *);
```

and using `my_compare` as the last parameter to `qsort`.

(See `sort-improved.c` on sample programs page.)

- (Other example use in *C In A Nutshell*.)

Some Interesting Things You Can Do In C

- Most UNIX/Linux “system calls” (requests to operating system) have a C library function to call them. Example — `fork` to create a new process. Most of them probably have a Java equivalent, but calling them directly from C may be interesting in being lower-level.
- Some functionality available in command shells is accessible via library functions — e.g., `readline` (tab completion, command history).
- The `ncurses` library provides functions to do fancier I/O (colors, cursor positioning, etc.).
- C “bindings” for OpenMP provide simple ways to do multithreading for performance.

Slide 4

Minute Essay

- None — sign in.

Slide 5