Slide 1

Administrivia

- Reminder: Homework 3 due today. I have office hours after 2pm if you need additional help.
- Reminder: Midterm exam next Tuesday. Review sheet on Web.
- I will grade today's quiz and — is it okay with everyone if I put your graded papers outside my office door for you to pick up?
- Watch your mail for grades and comments on homeworks turned in so far — coming soon. I will also post sample solutions on the Web, and return the written part of your Homework 1 in the same way as the quizzes.

Slide 2

Another Good(?) Quote

- From someone in a discussion group for the Java programming language: "Compilers aren't friendly to anybody. They are heartless nitpickers that enjoy telling you about all your mistakes. The best one can do is to satisfy their pedantry to keep them quiet :)"

Slide 3

Tips for Using `gcc`

- I mentioned last time that to (succesfully!) compile a program using the `sqrt` function you need the extra compiler flag `-lm`.
- Other flags that may be helpful:
  - `-Wall` to warn about things that are legit but probably/possibly not what you meant. Highly recommended!
  - `-pedantic` to warn about things that are non-standard. (Most compilers let you get away with some things that aren't strictly legit. But they don't all let you get away with the same things, so best to stick to what's "standard".)
  - `-std=c99` to accept C99 standard (default is the earlier one).
- Example use: `gcc -Wall -pedantic mypgm.c -lm`

Slide 4

Tips for Writing Shorter Code

- So far we've used variables whenever we wanted to calculate something. However, in some contexts we can just use an expression directly. For example, in the program to compute sum and product of two inputs, we could have omitted the variables `sum` and `product` and just written

```
printf("the sum of %d and %d is %d\n",
    input1, input2, input1 + input2);
printf("the product of %d and %d is %d\n",
    input1, input2, input1 * input2);
```

Slide 5

Functions and Scope

- In addition to having a type and a name, each variable has a *scope* in which it's valid. Variables declared inside a function can be used only within that function. Variables declared outside all functions can be used anywhere — *global variables*, almost always a bad idea.
- One result — variables with the same name in different functions are different variables.

Slide 6

Functions and Parameters

- We said last time that functions have *parameters.* Another word for them is *arguments* (you will see this is some compiler error messages). More terminology:
  - *Formal parameters* are the parameters as viewed from the function — can think of these as additional variables whose scope is the function.
  - *Actual parameters* are the values with which the function is called.
- When a function is called, actual parameters are copied to formal parameters — "pass by value", meaning that changes made in the function to its copies are not reflected in the calling program's copies.

Slide 7

Function Return Values

- Most functions return a value (but only one); it's the value of the expression following the keyword `return`, in the function definition. The type of this value is given as part of the function definition. If you don't want to return anything, can make this `void`. If you want to return two things? Later . . .
- Function calls are expression, so they have a value — whatever is returned by the function. Example of this last time, using `sqrt`.

  Another example — `scanf` returns a value that the calling program can use to tell whether the user entered sensible data. (Revise our make-change example one more time.)

Slide 8

A Little About the Midterm

- Review class notes, example programs from class, minute essays, quizzes, and homeworks. Focus more on those than on the textbook.
- Most questions will likely be more difficult (or at least longer) than quiz questions, but similar in format. Might be a few short-answer / multiple-choice questions too.
- Open book, open notes, but no access to Web.
- (Topic by topic through the review sheet.)

Slide 9

Minute Essay

- None — quiz.