Introduction to Computer Science
W 1113 – Lab (C)
Lab 6

Janak J Parekh
3/3/04

Recap from Lab 5

- Function prototypes
- Functions
- Conditionals
- Loops

2

Agenda

- Elements for HW#3
  - Variable scoping
  - Two-dimensional arrays
- Good coding practices
- Debugging
- Midterm review…

3

Variable scope

- Variables can be declared in different parts of your program, and this affects how they're accessible
- *Global* variables are declared outside any function
- *Local* variables are declared inside a function, or any arbitrary code block
- In C, local variables *must* be declared at the top of the block
- The "closest" one in the same block takes precedence

4

Example

```
#include<stdio.h>
int i = 5;
int main(void) {
    int i = 10;
    {
        int i = 12;
    }
    printf("%d\n", i);
}
```

- Yes, this is legitimate syntax! What's the answer?

5

A note on code blocks…

- Be *very* careful in identifying code blocks; use { } and proper indentation to keep your code clear
- If-else if-else: note that the latter two are *optional*, but should clearly correspond to the "original if" if present… legitimate syntax:

```
if(a) {
    if(b) { … }
    else { … }
} else { … }
```

6

Why global variables?

- If you have some piece of information used by lots of functions in the same program, no need to pass them as variables if they're already accessible
- However, be careful *not* to make everything global
- We'll get more used to structuring data later in the semester…

7

Permanent vs. temporary variables

- Book makes distinction – probably beyond the "scope" of this class
- Modern computers have a much larger stack
- Unless you're doing very special stuff, don't worry about it
- **static:** The most confusing keyword in C, ever

8

Two-dimensional arrays

- Easy to set up:
  - int a[10][20];
  - a[10][12] = 6;
  - Might want to "zero out" the array initially… how?
- Special meaning with strings
  - char strs[10][20];
  - You can treat this as a 2D array of chars, *or* as a 1D array of strings
  - In the latter, how many strings, and how many chars in each?
  - strcpy(strs[3], "Hello world");

9

Good coding practices

- Comment!
- Proper variable, function naming
  - In general, variables and functions have an initial lowercase, uppercase later
  - int numRecords = 0;
  - Indentation is very important, especially in keeping track of scope
    - emacs will help you in this
    - I've debugged people's code just by indenting it!

10

Good coding practices (II)

- Initial values for (most) variables
  - int i = 0;
  - int a[10] = { 0 };
  - Especially important in C – no presumed default
- Avoid very long functions: split up functionality
- Avoid overly complex logic if possible

11

Debugging tips

- gcc -Wall
  - Compile with "all warnings"
  - Often can catch errors this way
  - Sometimes will return some "optional" errors
- printf()
  - When stuck, print out intermediate results as your program runs

12

Using a debugger

- Especially with C code that crashes, it's hard to tell *why* the C code crashed
  - "Segmentation fault" isn't a very good answer
  - It'll only get worse when we learn pointers
- You can run your code through a debugger and see why it crashed
- Let's try a simple example…

13

Bad code

```
int main(void) {
  char c;
  strcpy(c, "This is a test");
}
```

- OK, this looks obvious here, but if you have a few hundred lines of code…
- Not surprisingly, it crashes

14

gdb – the GNU debugger

- First, compile your code with "-g"
  - gcc -g -o test test.c
- Then, run it with gdb
  - gdb test
- Common gdb commands
  - run
  - list – look at code
  - bt – "backtrace" along the function call stack
  - up/down – move among function call stack
  - break – add a "breakpoint"
- This is a whirlwind tour

15

gdb's unfriendly?

- Buy a commercial IDE
- Or, try ddd, which is a graphical frontend to gdb
  - Lots of features – I'll only scratch the surface in my "tour"
- You probably don't need to use a debugger for HW#3, but it'll be important for later homeworks

16

Midterm review…

- Any specific questions, first?
- Let's run through the slides

17