

ELE101 Precept 3

James Donald
February 19-20, 2004

Arrays

Setup Instructions

- Download the precept 3 files from Blackboard and unzip **precept3.zip** into a new folder preferably somewhere on your H:\ drive.
- Open up **ele101login.bat**
- Use the 'cd' command to reach the new folder where you unzipped the contents of **precept3.zip** from within the shell.
- *NEW*: This time, open up the emulator with the following command:
 - bash.202\$ **emulator_precept3.bat &**

Review & Outline

- Last time:
 - Writing, compiling, and testing our own programs
 - Selection (Ch. 5)
 - `if (a == 2) { /* .. */ } else { /* .. */ }`
 - Loops (Ch. 6)
 - `while (j > 0) { /* ... */ }`
 - `for (i = 0; i < 5; i += 1) { /* ... */ }`
- Today:
 - Arrays

What are arrays and do we really need them?

- Open up the file **Cmd-noarray.c** with your favorite editor.
- To test it, just type the word **noarray** into the ELE101 prompt in the emulator. This time the programs have been preloaded.

```
int Grade_of_Student_1;  
int Grade_of_Student_2;  
int Grade_of_Student_3;  
int Grade_of_Student_4;  
int Grade_of_Student_5;
```

	90	91	92	93	94	
--	----	----	----	----	----	--

- Our program works, but it's a bit clunky isn't it? Imagine if we tried to scale it to handle a class of 100 students.

Setting up arrays

- Take a look at **Cmd-array.c**
- Apparently much more concise. Let's try running it to make sure it has the same behavior.
- Declaring and assigning arrays:

```
int Grade_of_Student[5] = { 90, 91, 92, 93, 94 };
```
- OR this can be written as:

```
int Grade_of_Student[5];  
Grade_of_Student[0] = 90;  
Grade_of_Student[1] = 91;  
Grade_of_Student[2] = 92;  
Grade_of_Student[3] = 93;  
Grade_of_Student[4] = 94;
```



Accessing arrays

- What if we wanted to know the grade of only student #2?

```
printf("#2 got %d points.\n",  
      Grade_of_Student[1]);
```
- How to make some part of the program decide to give a C- to student #3?

```
Grade_of_Student[2] = 70;
```
- How to print the entire array?

```
for (i = 0; i < 5; i++) {  
    printf("%d\n", Grade_of_Student[i]);  
}
```

Other tips

- Be careful not to try to access an array outside of its bounds. What do you see wrong with the following piece of code?

```
for (i = 1; i <= 5; i++) {  
    printf("The grade of student %d is \  
          %d\n", i, Grade_of_Student[i] );  
}
```
- Multi-dimensional arrays
 - Won't cover this in detail today, but you can take a look at **Cmd-mdarray.c** for a Tic Tac Toe example.

Administrivia

- Homework assignment #2 due Tuesday Feb 24th
- Midterm on Thursday March 4th
- Done with homework assignment #2 already? Great job, and you don't need to stay for the next 20 minutes of precept.
- Article from Wednesday:

CNN.com

[Google adds 1 billion pages to search](#)

Google and Yahoo! are revving up their online search engines as the once-friendly business partners shift gears and prepare for what promises to be a fierce rivalry.



Review on bases

- Pop Quiz.
 - Convert 10101_2 into base 10.

$$1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 21_{10}$$

- Convert $2D_{16}$ into base 10,

$$2 \cdot 16^1 + 13 \cdot 16^0 = 45_{10}$$

Base conversion example

- Let's convert 62_{10} into base 7
- In base 7, our available digits are: 0123456.

$$62 / 7^2 = 1 \text{ remainder } 13$$

$$13 / 7^1 = 1 \text{ remainder } 6$$

$$6 / 7^0 = 6 \text{ remainder } 0$$

- Final answer: $62_{10} = 116_7$

Sample Base-7 Conversion Program

```
int remainder, digit, power;
/* ... */
remainder = StrAToI(str);
power = 343; /* 7^3 */
while (power > 0) {
    digit = remainder / power;
    printf("%d", digit);
    remainder = remainder % power;
    power = power / 7;
}

```

The full program is available as `Cmd-base7.c`. Let's test it.

Base conversion: what else do we need?

- Problems that you would need to fix with the code fragment on the previous slide:
 - Can't handle numbers greater than or equal to 7^4 (2401_{10})
 - If you enter a small number you get preceding zeroes in the output, e.g. $13_{10} \rightarrow 0016_7$
 - And what about overflow?
- But there are also some completely different ways of doing the conversion. Feel free to be creative.
- On top of that, for your homework assignment you'll need to implement support for bases 2-36 and error checking.
- Good luck!