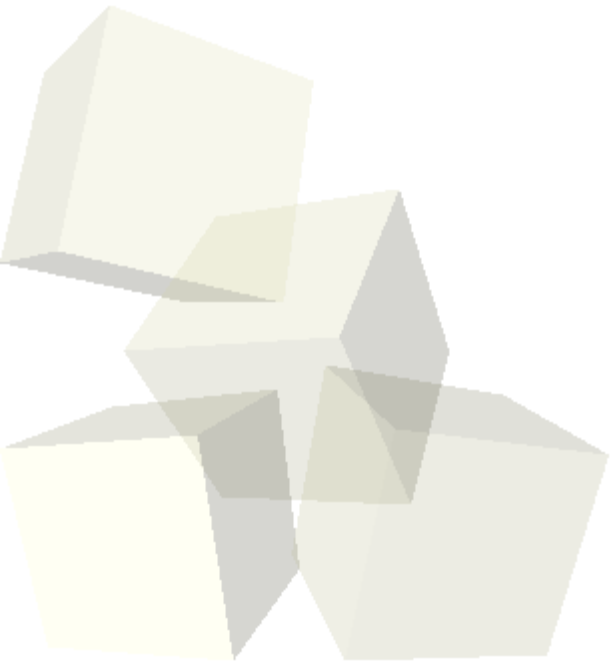




Introduction to PAD1

Dr. Mark Lewis

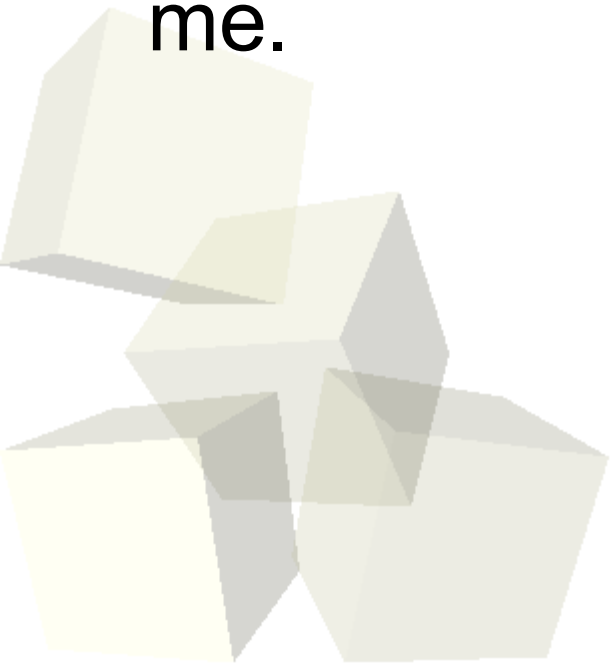
8-24-2006





Opening Discussion

- I will start off every class with a little discussion. Typically this discussion includes questions about the previous class.
- Today I want to use this time to get to know a bit about you.
- I'd also like to give you a brief introduction to me so you can know some of the things that motivate me.

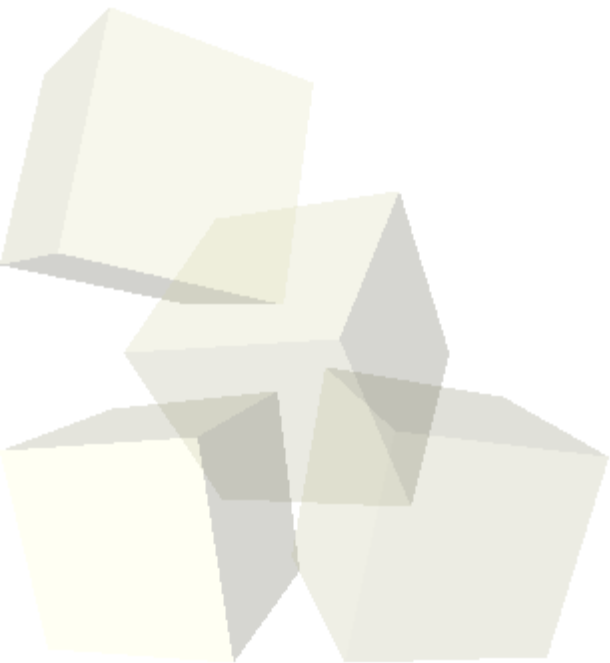




- Course web site:
www.cs.trinity.edu/~mlewis/CSCI1320-F06/
- Office: HAS 201K
- Office hours: 9:30-11:00am TR. Open labs 4:30-6:00pm T and 2:00-6:00 W.
- Phone: 7022
- E-mail: mlewis@trinity.edu (This is the best way to reach me most of the time.)
- There is a schedule on the web site listing all topics and when things are due. There are also links to my lecture notes.



- We will use “Computer Science: A Structured Programming Approach Using C”.
- You will be expected to do readings from this book and we will be following it rather closely.



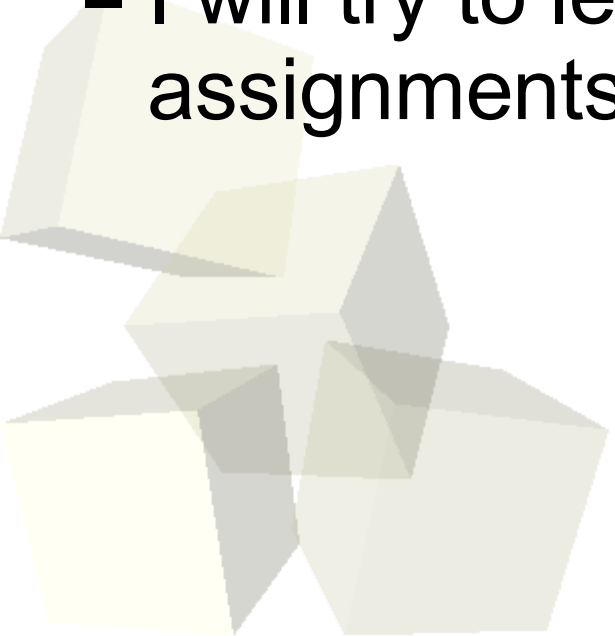


Course Description

- This course provides an introduction to programming with a primary focus on problem solving and logic.
- The course will specifically teach you how to program in the C programming language. It will also teach you to work in Linux in a command-line environment and with basic text editors.
- My courses tend to be somewhat rigorous. My overriding objective is to make you think. Because of the nature of this course I want to go beyond just giving you new things to think about, but instead give you tools to think about things in new ways.



- Writing great programs is more art than science.
- Computers are the ultimate creative medium and also the ultimate creative outlet. The way we interact with them seems formal to some, but there can be a lot of fun in just looking for ideal ways to express ideas in the formal systems of different computer languages.
- I will try to let you express creativity in the assignments for this class.





Assignments

- The largest chunk of your grade in this class is determined by the 10 assignments that you will do over the course of the semester.
- These are mostly small problems that you will solve by writing code, but some sets of assignments will link together allowing you to produce a larger product.
- You can work with others on design and get help on specific problems, but what you turn in must be your own construction. Everything you turn in for a grade is pledged if you are under the honor code.

- Your grade comes from four different components.
 - ◆ Assignments (10) – 50%
 - ◆ Tests (2) – 30%
 - ◆ Quizzes (6 with lowest dropped) – 10%
 - ◆ Class Participation – 10%
- The midterm and final are both equally weighted.
- The quiz questions are modeled after test questions so you will have an idea of the style to expect on the test.
- There are quite a few different opportunities for extra credit.
- Class participation includes attendance and your actual participation during class.



Organization of Class

- I have yet to find the perfect way to teach these classes. I'm always torn between the need to include all the content that is needed for later CS courses and having you apply as many ideas as possible so you will retain the information.
- The approach I'm taking this year is to split topics between classes. I will introduce a topic at the end of one class. I expect you to read about it before the next class. During that following class we will spend a lot of time applying what was in the reading.
- This active format should help you actually learn the material, but only if you read.



- We will be using Linux for this class and programs will be written in ANSI standard C.
- I will teach you how to get around in Linux with the command line as well as how to compile and run programs. Your programs must compile and run on these machines with GCC.
- You should get an e-mail giving you your Linux password before next class. That way you can all follow along with what I'm doing then. You will be logging in most days we meet.
- There is a PDF with a short intro to Linux and vi on the website.

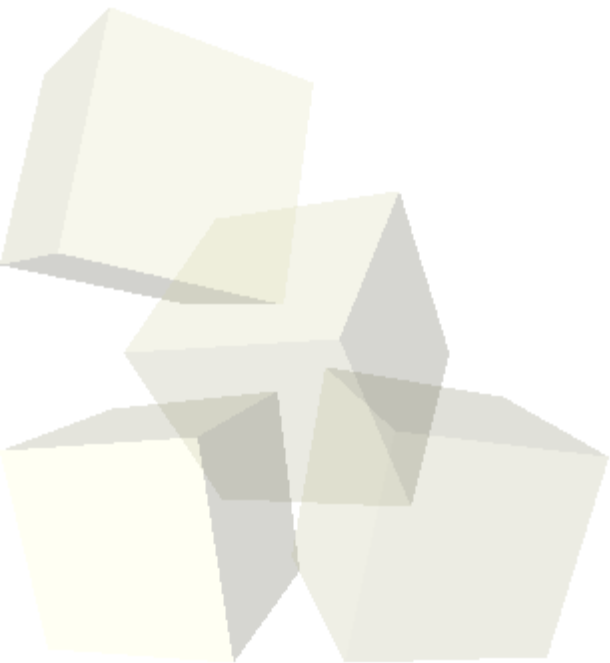


- Term used to be a position that humans could hold.
- They facilitate/dominate every facet of your life, even when you don't realize it.
- All modern computers use some derivative of the von Neumann architecture. This is a fetch, execute, store setup with a single bus.
- The languages we use to communicate with these machines has evolved over time.
- We are currently seeing a massive move from serial to parallel computation.



Integers and Bases

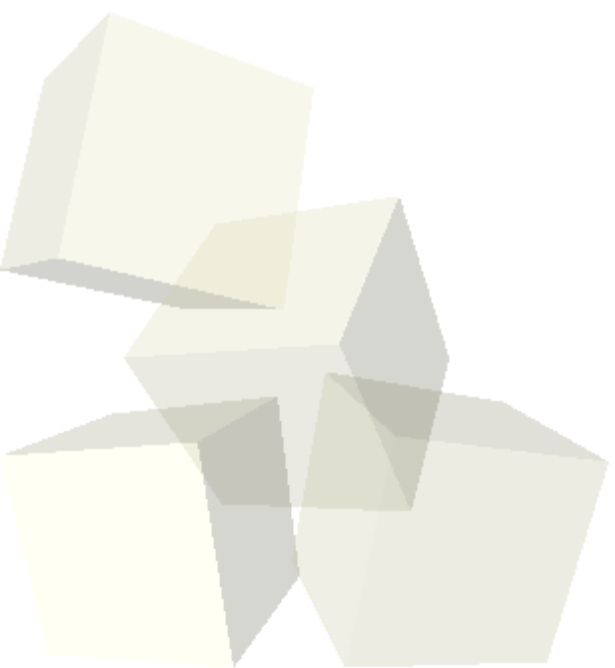
- At a certain level all that computers do is manipulate symbols. Typically these symbols are numbers stored in binary format.
- The advantage of the computer is that it can manipulate these numbers with blazing speed.
- Let's take a minute to review the concept of integers and different bases.





Binary Numbers

- Binary numbers are simply numbers in base two. There is nothing special about them except that they are represented by the digits 0 and 1 alone.
- How would you convert from binary to decimal and back? Let's do a few examples.





- This class is about basic programming. C is a fairly simple language that doesn't obfuscate our objective.
- Unlike many “teaching” languages, which are often intentionally simple, C is actually used in the real world.
- What is more, many modern languages use C as a foundation. These “C family languages” are some of the most popular in the world and all use a syntax similar to C. If you can program in C you can read and understand most of these languages easily.



A Simple C Program

- I'm going to write a simple C program at this point. You aren't expected to understand what I'm doing. The reading will help with that, as will the fact that you will follow along when we do this next class. I just want to give you a sample of what C code looks like.
- The further we go with C the more you will notice it is a very terse language. Many aspects were designed to minimize the number of keystrokes. This was critical at the time C was created. It has likely contributed to the success of C family languages as feeding on the inherent laziness of programmers.



- I will end every class by asking you to write a few sentences on one or more questions. I also use this to keep attendance so make sure your names are on what you give me.
- Do you have any questions about the material presented today?
- What do you want to get out of this course?
- Remember to read chapters 1, 2, and Appendix D.1 before next class.

