

Today's Topics

**Computer Science 1
Review**

Upcoming

Final Exam: Saturday, 12/13, 2:00pm, B101 LSRC

Review Session: TBA

Reading

***Great Ideas*, Chapter 4 - 15**

4. Top-Down Programming, Subroutines, and a Database Application

- **Functions using Functions**
- **Getting Information In and Out of Functions**
- **Class Data: known within class.**
- **Formal Parameters/Arguments**
- **Syntax: Using a Function**
- **Functions that Return Values**
- **Syntax: Defining a Function**
- **Larger Problems: How to Deal with the Complexity**
 - **Divide and Conquer**
 - **Design: Stepwise Refinement**
 - **Top-Down Implementation**

4. Top-Down Programming, Subroutines, and a Database Application

- **"Parallel" Arrays or "Corresponding" Arrays**
 - **Model Phone Book Capability**
 - **Typical Access by Name**
 - **Access by other Fields (other arrays)**
- **Extend Idea to Database**
- **Basic Data Base Functions**
- **Wild Card Retrieval**
- **Used Car Database**
- **Relational Data Bases**

4. Top-Down Programming, Subroutines, and a Database Application

- **Recursion**
 - **Factorial (N!)**
 - **Iterative Approach for Factorial**
 - **Exponentiation (X^N)**
- **Church-Markov-Turing Thesis**
 - **This part of Java lets you solve all kinds of algorithms**

5. Graphics, Classes, and Objects

- **Basic Stuff**
 - **Canvas class, Graphics class, pixels, Coordinates**
- **Graphics Methods**
 - `void drawLine(int x1, int y1, int x2, int y2)`
 - `void drawRect(int x, int y, int width, int height)`
 - `void drawOval(int x, int y, int width, int height)`
 - `void setColor(Color c)`
- **Example: (Using Recursion) Serpinsky.java**

5. Graphics, Classes, and Objects

- **Writing a Class**
 - **Header**
 - **Contents of a class definition**
 - **The Constructor**
 - **The `serp` Class to draw Sierpinsky Gasket**
- **Simple-Minded Animation**
 - **Draw and Erase**

6. Simulation

- **Simulation: Motivation**
- **Optimization, Simulation: Biggest Dog Lot**
- **How Could We Automate Process?**
- **Other Roles For Simulations**
 - **Economy, Policy (e.g. birth control), Marketing**
 - **Camera Lenses, UNC CS Walkthrough, Virtual Reality**
- **Simulation in Microelectronics**
 - **Logic, Layout, Circuit, Process**
- **Design and Manufacturing**

7. Software Engineering

- **Engineering a Program - Programming in the Large**
- **What is Good Software?**
- **Program Life Cycle, Feedback Cycles**
- **Understanding Problem / Specifications**
- **Debugging**
- **Correctness, Proofs?**
- **Documentation**
- **Testing**
- **Bottom Line: Productivity: 15 LINES OF CODE/DAY**
- **Many People? The "Committee": Interaction**
- **Organizational Schemes: e.g. *Chief Programmer Team***

7. Software Engineering

- **Killer Robot Scenario**
 - **Development Models**
 - **Waterfall**
 - **Prototyping**
 - **Testing**
 - **User Interface**
- **Ethics**

8. Machine Architecture

- **Architecture (definition)**
- **Hardware / Software**
- **Basic Computer very primitive**
- **Architectural Features**
 - **Memory**
 - **CPU: AX, IP, IR, CF**
- **Fetch/Execute Cycles**
- **Need to handle IF and WHILE situations**
- **Tracing (often the only way to understand)**
- **Loop Example: Factorial Example**
- **Handling Lists or Arrays (Self Modifying Code)**
- **Fancier Architecture**

9. Language Translation

- Importance of language
- Goal: *Translate Java To Assembler*
- Revise Syntactic Production Rules (seen before)
- Use Rules to Modify Strings
- Add Semantic ("meaning") Components to our Rules
- Use Syntactic Derivation to Generate Semantic Rules;
Use Semantic rules to Generate Code
- Rules for Looping
- Important: *Everything done by simple substitution*
- Everything "adds up"

Electric Circuits

- **Levels of a Computer System**
- **Circuits: Water Model (the real thing = electrons)**
 - **battery, generators, heat -> light, motors**
- **Circuits With Switches (e.g. knife switch)**
- **Logic/Truth Tables: AND, OR**
- **Implementing Logic with Switches**
- **Logical (Boolean) Expression**
- **Equivalence of:**
 - **Circuit with Switches, Truth Tables, Boolean Expression**
- **Arbitrary Truth table for $f(x,y,z)$**

Electric Circuits

- **Relays**
- **Storing Information (Memory): Latch**
- **Binary Numbers**
 - **Conversion to and from Decimal**
- **Binary Addition**
 - **Truth Tables**
 - **Block Diagram**
 - **Simple *Adder* Circuit**
 - **Decoding/Control**

12. Computer Communications

- **Computer Communications is one of the Great Ideas**
- **Modes of Communications**
- **Like Most of Computing: Layers upon Layers**
- **Basic Communications: In binary**
- **Connection Mode**
 - **Circuit Switched, Message Switched, Packet Switched**

TCP/IP

- **Ethernet (Bus Example)**
- **Internet -- a network of LANs that are interconnected**
- **Packets -- the currency of the Internet**
- **The Layers**
 - **The Physical Layer, The IP (Internet Protocol) Layer**
 - **The TCP Layer, The Application Layer**

12. Computer Communications

- **Packets Within Packets (Encapsulation)**
- **Reliability**
- **Addressing (Layers Again!)**
 - **Hardware Address (Ethernet Address)**
 - **IP Address**
 - **Domain Name (address)**
- **Applications**
 - **email, news, talk, ftp, telnet, ssh, rlogin**
 - **information services: WWW, Older: gopher, WAIS**
- **Client/Server**
 - **Print Server, File Server, Name Server,**
 - **WWW**

11. Security, Privacy and Wishful Thinking

- **Billions in Losses**
- **Possible Traps in Public Systems**
 - **Trojan Horse, Onlooker, Digital camera**
- **Good Passwords and Cracking**
 - **Briefcase combination lock**
 - **Analysis of brute force methods**
 - **Password on a Computer**
 - **Dictionary Attacks**
- **Encryption**
 - **Monoalphabetic Substitution**
 - **Polyalphabetic Substitution**
 - **The Vignere Cypher; The Babbit Solution**

11. Security, Privacy and Wishful Thinking

- **Cypher Reuse: BAD**
- **One Time Pads: Can be Absolutely Secure**
- **The Key Exchange Problem**
 - **Using your "secure" channel (bad)**
 - **A Padlock Analogy**
- **Public Key Encryption**
 - **A Padlock Analogy**
 - **Rivest, Shamir, and Adleman (RSA) Encryption**
 - **Using Public Key and Private Key**
 - **Primes and Factoring**
 - **Breaking the Code: *Factoring***

11. Security, Privacy and Wishful Thinking

- **Public Key Encryption**
 - **Digital Signatures**
 - **Using Private Key and Public Key**
 - **Need for Time Stamps**
- **Other Attacks (Buzz Words)**
 - **Many Leave No Trace**
 - **Password Hacking, IP Spoofing, Replay Attack**
 - **Man in the Middle, Denial of Service**
- **Whom Can You Trust?**
 - **Viruses, Trapdoors, Trojan Horses, Common Sense**
- **The Strong Encryption Trap**

10. Virtual Environments for Computing

- **The Raw Machine Provides a Hostile Environment**
- **Early Years Had Major Theme: CPU Time Precious**
- **Later Years: Cheaper and Cheaper Hardware**
- **What Does an Operating System Do?**
 - **Processor Management (Multiprogramming)**
 - **I/O Systems**
 - **Memory Management**
 - **Software Environments**
- **Memory Management**
 - **Memory Hierarchies, Paging, Protection**

10. Virtual Environments for Computing

- **I/O Systems**
 - **Files Systems, Communications/Networking**
 - **Graphical User Interfaces (GUI)**
- **Processor Management**
 - **True Parallel Processes vs Simulated**
 - **Synchronization**
 - **Race condition**
 - **Deadlock**

Changing Computer Technology

- **Some Fundamental Limitations**
 - **Speed of light, heat dissipations, capacitance and inductance**
- **Other Important Concerns**
 - **Economics !!!, Noise, Lifetime (mtf), Space**
- **Relay Computers (and problems)**
- **Vacuum Tube Computers (and problems)**
- **Transistor**
- **Integrated Circuits -- VLSI**
- **Economics of Silicon (Micro-electronics): CPUs in Everything**
- **Technology Summary (table)**

13. Program Execution Time

- **On the Limitations of Computer Science**
 - 1. too slow. 2. Non-computable. 3. Don't know algorithm
- **Time Complexity, N**
- **Study of a Sorting Algorithm: Selection Sort: N^2**
- **Polynomial = Tractable**
 - **Linear Time Algorithms: $t = A * N$**
 - **Cubic Time Algorithms: $t = A * N^3$**
 - **Quicksort: $t = A * N * \log(N)$**
 - **Binary Search: $t = A * \log(N)$**
- **Intractable Algorithms: Exponential $t = A * B^N$**
 - **Chess, Traveling Salesperson, Towers of Hanoi**
- **More hardware not always the answer!**

14. Parallel Computation

- **Limitation on Processor Speed**
 - **Speed of Light**
 - **Manufacturing Problems with Small Sizes**
 - **Heat Dissipation**
- **Ultimately Parallelism is Only Hope**
- **Forms of Parallelism**
 - **Word Size, Pipe Line (Laundry Example)**
 - **Multiprocessors, Networks of Processors, Internet**
- **Speedup**
- **What can we do with 100 processors?**
 - **Even with optimal speedup no big help for B^N programs**

15. Noncomputability

- **Certain Problems Not Amenable to Computer Solution**
- **Existence of Noncomputable Functions**
 - **Approach: *Matching up Programs and Functions***
 - **Have: Uncountable Infinity of Functions (cannot be put into a row)**
 - **All Programs Can be Ordered**
 - **Try to Draw Lines Between Functions and Programs**
 - **Many *more* Functions than Programs!**
- **Programs that Read Programs**
 - **E.g., Java Compiler**
- **Solving the *Halting Problem***

15. Noncomputability

- **Proofs by Contradiction (Indirect Proof)**
- **Proving non-computability**
 - **Assume Existence of Function `halt`:**
 - **Use in way resulting in *Paradox!***
 - **Therefore `halt` cannot exist!**
- **What Does It All Mean?**