Computer Science: An Overview
Syllabus

2013.9.18

Schedule:

Chapter	Time Scheduled	Date
1. Data Storage	4 * 45’	9/18, 9/25
2. Data Manipulation	3 * 45’	10/9, 10/16
3. Operating Systems	3 * 45’	10/16, 10/23
4. Networking and the Internet		
Quiz	1 * 45’	10/30
5. Algorithms	3 * 45’	10/30, 11/6
6. Programming Languages	3 * 45’	11/13, 11/20
8. Data Abstraction	3 * 45’	11/20, 11/27
9. Database Systems (File Structure)	2 * 45’	12/4
11. Artificial Intelligence	4 * 45’	12/11, 12/18
12. Theory of Computation		
Review	2 * 45’	12/25
Final Examination	2 * 45’	TBD

Legend:

- Reading Required
    - ■ Know **how** to do this
- *Introduced in class but not included in homework/exam*
- ~~Included in textbook but will not discussed in class~~

121 Pages, 37 Key Points

Note: Page numbers and chapter titles are based on the 11th edition of *Computer Science: An Overview*.

Chapter 1 Data Storage (4 * 45’) (16 pages)

1.1 Storage of Bits (p.20 - p.24)

- Use Boolean operations to analyze logic circuits
- Understand what is a logic gate. Figure out the external properties of a given flip-flop circuit.

1.2, 1.3 Main Memory and Mass Storage (p.26 - p.35)

- Know the storage formats of disks and CDs.

1.4 Representing Information as Bit Patterns (p.35 - p.40)

- Know that we can represent all kind of information as bit patterns

1.5 The Binary System (p.42 – p.46)

- Convert integers and decimals from base ten form to binary form, and vice versa.

1.6 Storing Integers (p.47 – p.52)

- Convert integers to two's complement notation, or vice versa.

1.7 Storing Fractions (p.53 – p.57)

- Encode decimals into floating-point format, and decode bit patterns in the format. (Convert tables of excess notation will be given.)

1.8 Data Compression (p.58 – p.60)

- Know what lossy compression and lossless compression are.
- Know the basic idea of Huffman Code and Run-length encoding

Chapter 2 Data Manipulation (3 * 45') (20 pages)

2.1 The Von Neumann Architecture (p. 74 – p.76)

- Understand the 3 characteristics of the Von Neumann Architecture and the related concepts.
- The Stored-Program Concept

2.2 Machine Language (p.77 – p.82)

- Be familiar with V8 and its instruction set.

2.3, 2.4 Program Execution / Arithmetic/Logic Instructions (p.83 – p.93)

- 'Execute' programs in the machine language of V8. (A simplified version of V8 will be given.)

Chapter 3 Operating Systems~~ and Networks~~ (3 * 45') (6 pages)

3.1 The History of Operating Systems (p.110 – p.114)

- Know the development of OS for single processor systems

3.2 Operating System Architecture (p.119 – p.121)

- Know the components of an operating system.
- Know how an operating system gets started.

3.3 Coordinate the Machine's Activities (p.122 – p.124)

- Understand the concept of a process and the difference between a program and a process.
- Know what process switch is.

3.4 Handling Competition (p.125 – p.129)

- Know that an important task of an OS is the allocation of resource

- *What is deadlock*

Chapter 4 Networking and the Internet

4.4 Internet Protocols (p.167 - 173)

- *Network protocol layers*
- Know what IP address is.
- Know how packets are routed.

Chapter 5 Algorithms (3 * 45') (31 pages)

5.1 The Concept of an Algorithm (p.188 - p.190)

- Know the 4 keywords of algorithms.
- Understand the difference among algorithm, program and process.

5.2 Algorithm Representation (p.191 - p.198)

- Know that there are ways to represent algorithm in mind
- Read/Write algorithms in the pseudocode

5.3 *Algorithm Discovery (p.198 - p.203)*

- *Know that it's a challenging ART*

5.4, 5.5 Iterative and Recursive Structures (p.204 - p.212, p.214 - 221)

- Design algorithms in both iterative and recursive structures for simple problems.
- Follow an iterative or recursive algorithm step by step.

5.6 Efficiency (p.222 - p.226) ~~and Correctness~~ (Refer to Section 9.5)

12.5 Complexity of Problems (p.527 - p.535)

- Figure out the time complexity of a given algorithm.
- Know the problem classification (polynomial problems, nonpolynomial problems, NP problems and NP-complete problems).

Chapter 6 Programming Languages (3 * 45') (8 pages)

6.1 Historical Perspective (p.240 - p.248)

- Know generations and paradigms of programming languages

6.2 Traditional Programming Concepts (p.248 - p.260)

- Know the concept of structured programming

6.3 Procedural Units (p.260 - 267)

- Know what global variables and local variables are.
- Determine the results of executing a procedure whose parameters are passed by value or by reference.

6.4 Language Implementation (p.268 - p.275)

- Know the translation process of languages.
- Figure out the expression from a parse tree.

Chapter 8 Data Abstractions (3 * 45') (17 pages)

8.1 Basic Data Structures (p.342 - p.344)

8.2 Related Concepts

8.3 Implementing Data Structures (p.348 - p.361)

8.3.1 Arrays

- Determine the address polynomial for a multiple-dimensional array.
- Insert, delete or search an entry in an array.

8.3.2 Lists

- Know the implementation of contiguous lists and linked lists.
- Insert, delete or search an entry in a linked list.

8.3.3 Stacks

- Know the implementation of stacks.
- Push and pop entries; test for an empty or full stack.

8.3.4 Queues

- Know the implementation of circular queues.
- Insert or remove entries from the queue, as well as detect whether the queue is empty or full.

8.3.5 Binary Trees

- Know how binary trees are stored with or without pointers.
- Find any node of the tree.

Chapter 9 Database Systems (2 * 45') (8 pages)

9.5 Traditional File Structures (p.406 - p.413)

- Know that OS maintains the PL access request to files and the concept of file descriptor table
- Know how sequential files are stored in mass storage
- Know how to detect EOF when traversing a file
- Merge 2 sorted files
- Know the structure of indexes for direct access (relationship among index files and indexed files).
- Know how data entries are dispersed among buckets.
- Know the division hash function is with distribution problems.

Chapter 11 Artificial Intelligence (2 * 45’) (11 pages)

11.1 Intelligence and Machines (p.462 – p.466)

- Know what’s Turing Test

11.3 Reasoning (p.473 – p.483)

- Build a search tree for an eight-puzzle either in the breadth-first manner or using a given heuristic.

Chapter 12 Theory of Computation (2 * 45’) (4 pages)

12.2 Turing Machines (p.512 - p.515)

- Determine the final configuration of a given Turing Machine.

12.3 Universal Programming Language (p.516 – p.521)

12.4 A Noncomputable Function (p.522 – p.526)

- Know what unsolvable problems are and that the halting problem is unsolvable.