1. Data storage (including number representations)
    a. Bits and their storage
        i. Boolean (And, OR, and Xor)
        ii. Gates and flip-flops
            1. Flip-flip = (hold value after input changes to 0)
        iii. Hexadecimal notation
            1. 0 to F
    b. Main memory
        i. Memory organization
            1. High order bit, → low order bit
        ii. Measuring memory capacity
            1. Power of two since its more convenient
            2. KB = 1000 and kibi B = 1024 etc
    c. Mass Storage
        i. Magnetic systems
            1. Normal hard drives
                a.
            2. Storing information on tape
                a. Turning
        ii. Optical systems
            1. CD, DVD etc.
                a. External systems
        iii. Flash based systems
            1. SSD
                a. Super fast compared to other mass storage systems
            2. SD
                a. Camera cards and the likes
    d. Representing information as bit patters.
        i. Representing text is bit patterns using ASCII
            1. Maybe write some code as an ensample or something.
        ii. Representing numeric values
            1. 0000 = 0
            2. 0001 = 1
            3. 0010 = 2
            4. 0011 = 3
        iii. Representing images
            1. Bit map 0 to 255 (show example from/in Matlab if needed)
        iv. Representing Audio
            1. Store amplitude of wave and save it at a interval.

e. The binary system

i. Binary system

1. Base 10 vs base 2

ii. Binary addition

1. 01010101 + 10101010 = 11111111

2. 00100110 + 10100010 = 11001100

iii. Fractions in binary

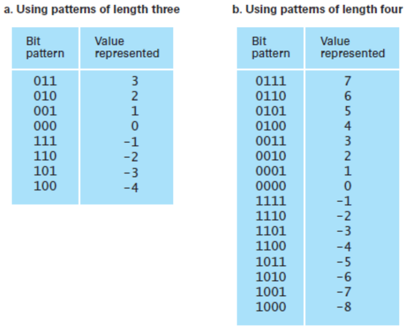
1. From book

a. 101.101 = 5.625

i. 4+1+(1/2)+(1/8) = 5.625

f. Storing integers

i. Two's complement notation



a. Using patterns of length three

Bit pattern	Value represented
011	3
010	2
001	1
000	0
111	−1
110	−2
101	−3
100	−4

b. Using patterns of length four

Bit pattern	Value represented
0111	7
0110	6
0101	5
0100	4
0011	3
0010	2
0001	1
0000	0
1111	−1
1110	−2
1101	−3
1100	−4
1011	−5
1010	−6
1001	−7
1000	−8

1.

2. The problem with overflow

a. 1011 + 0111 = 0000

b. 11 + 7 = 0

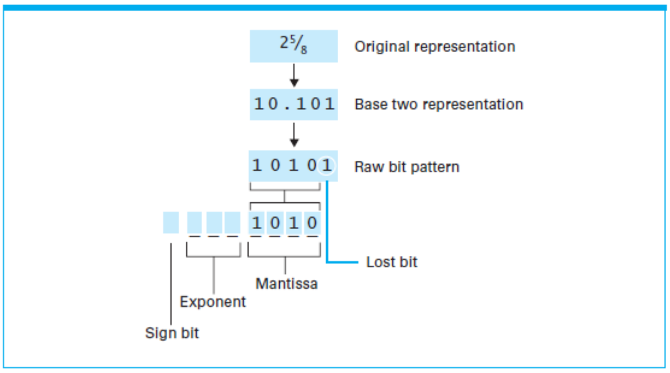
ii. Excess notation

1. Subtract value of highest order bit

g. Storing fractions

i. Floating point notation

1. From book



a.

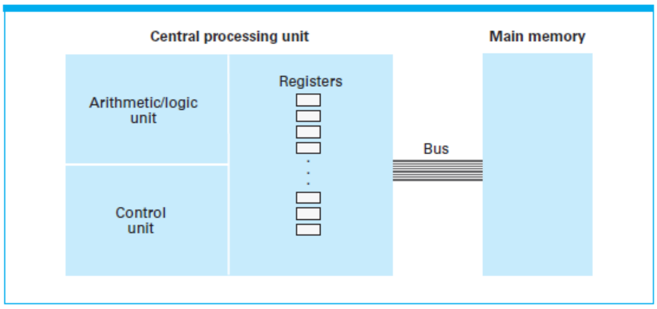
2. From Joan

a. Write here

ii. Truncation errors

1. When information is lost like above

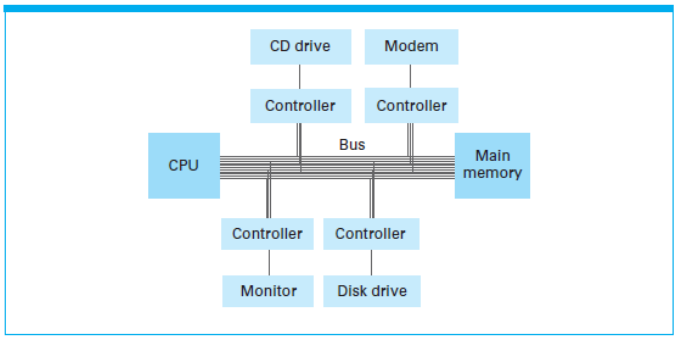
h. Data and programming
    i. Python
        1. Print “hallo world” etc
    ii. Variables
    iii. Operators and expressions
    iv. Debugging
i. Data compression
    i. Lossless vs lossy
        1. Same data/loose some data
    ii. Compressing images
        1. Integer division on the whole bitmap and to get lower values and multiply it again to get the image bad (lossy)
    iii. Compression Audio
        1. Removing information that the human ear can’t hear
        2. Integer division on the whole amplitude map and multiply again to get the audio back (lossy)
j. Communication errors
    i. Parity bits
        1. First bit in a in byte in example, when there is a even number of 1s its 1 and when its uneven its 0.
    ii. ECC (Error Correcting code)
        1. Every symbol has a code and its picks the ones that is the shortest when it detects an error.

2. Data manipulation (machine architecture)
    a. Computer architecture
        i. Cpu basis
            1. 

        ii. The stored program
            1. In older days one system could only run program one, since the hardware was specifically designed to run it. It was first later that it was found out that a system could run multiple different program by storing the program in the main memory, and that you wouldn't need to rewire the whole cpu.
    b. Machine language
        i. The instruction repertoire
            1. RISC (Reduced Instruction set computer)
            2. CISC(Complex instruction set computer)
                a. Used is most modern computers
        ii. An illustrative machine language
            1. Appendix c
                a. No fucking clue atm
    c. Program execution
        i. The program is loaded from the main memory into the register. From there its run by the processor.
        ii. The CPU preforms its job by running a tree step process known as the machine cycle. The steps in the cycle is fetch decode and execute.
        iii. An example of a program execution
            1. See book page 106 to 107
        iv. Program versus data
            1. Many programs can be stored in the main memory at the same time and the processor has no way of knowing if its data or a program its dealing with, since its all 1s and 0s.
    d. Arithmetic/logic instructions
        i. Logic operations
            1. And, OR, Xor
        ii. Rotating and shifting operations
            1. Rotating is rotating the whole bit string so the one at the end goes to the start or at the start to the end.
            2. Shift means shifts the whole but string by one and replaces the first one with either 0, 1 or what got pushed out at the end(rotate)
        iii. Arithmetic operations
            1. Things not supported by the cpu? Like floating point etc
    e. Communicating with other devices

i. Controllers

1.



ii. Direct memory access

1. Since a controller is connected directly to the computer's memory bus it can read data from a disk drive while processor preforms a process.

iii. Handshaking

1. Two way communication so two devices can share status and coordinates their activates.

iv. Popular communication Media

1. Parallel communication

a. Data transfer over multiple different lines

2. Serial Communication

a. Data transfer over a single line

3. Communication rates

a. Speed

i. Normally in bits per sec

b. Multiplexing

i. Data transfer both ways at the same time.

f. Programming data manipulation

i. Logic and shift operations

1. 5 xor 4 is 1

2. 5 or 4 is 5

3. 5 and 4 which is 4

ii. Control structures

1. If, while, etc,

2. Functions

iii. Input and output

1. Input = input from user

2. Output = output to user or from a function

g. Other architectures

i. Pipelining

1. Allowing the steps to be done overlappingly

a. So the next step is ready when the current one is done

ii. Multiprocessor machines

1. Parallel processing

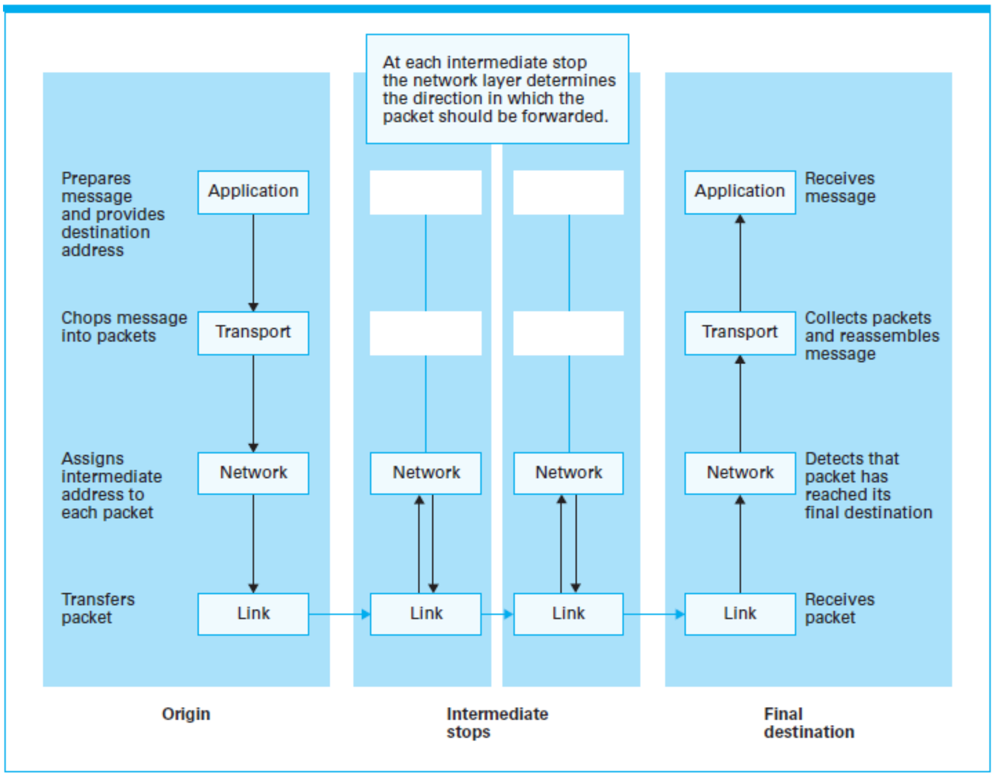
a. Multiple instructions can be run at the same time for a faster processing time

b. MIMD(Multiple-instruction stream, Multiple-data Steam) as oppse to SDMD (Single-instruction stream, Single-data Steam)

3. Operating systems
    a. The History of Operating systems
        i.
    b. Operating system architecture
        i.
    c. Coordinating the machines activities
    d. Handing competition among processes
    e. Security

4. Networking and the Internet
    a. Network fundamentals
        i. Network classifications
            1. Pan (personal area network)
                a. Normally within a few meters, bluetooth etc
            2. LAN (local area network)
                a. Within a network, home, school or workplace
            3. Man (metropolitan area network)
                a. Spanning a local community
            4. WAN (Wide Area network)
                a. World wide network ?
            5. AP (acces point)
                a. Acess to the network, its if a wifi ap, router or access from outside the network into it.
        ii. Protocols
            1. CSMA/CD (Carrier sense, multiple access with collision detection)
                a. Device waits until network is quiet to say something and if multiple both say something both stop and try again
            2. CSMA/CA (Carrier sense, multiple access with collision avoidance)
                a. Request to send/clear to send
        iii. Combining networks
            1. Repeater, switch, bridge and routers
                a. You know these of the top of your head.
                b. Gateway, forewarding table, DHCP etc
        iv. Methods of process Communications
            1. Client – server
                a. FTP
                b. Websites
                c. Etc of the top of your head
            2. Client – Client
                a. P2p
                    i. Torrents
                    ii. Games
                    iii. Some communication services
            3. Server – Server
                a. SMTP
        v. Distributed systems
            1. Cluster computing
                a. Handing data in a large data center
            2. Grid computing
                a. Lots of server spread all over the globe to handle request locally
            3. Cloud computing
                a. You can rent more servers on demand
    b. The Internet
        i. Internet architecture
            1. ISP (Internet service provider)

            2. Tier-1
                a. Backbone
            3. Tier-2
                a. Regional
            4. Tier-3 (intranet, Independent network)
                a. Access ISP
                    i. The ones who provide internet to consumers
        ii. Internet addressing
            1. Ip addresses
                a. IPv4
                    i. 32 bit = 2^32 Binary
                b. IPv6
                    i. 128 bit = 2^128 Hexadecimal
                c. Ips are assigned by the "Internet corporation for assigned names and numbers(ICANN)
                d. Top level domains
                    i. .com, .dk .ing. etc
                    ii. For countries they are called Country code TLD
                    iii. Domain name systems (DNS)
        iii. Internet applications
            1. In older times there for protocols for everything
                a. Network news transfer protocol(NNTP)
                b. File transfer protocol(FTP)
                c. Telnet and later SSH (secure shell)
            2. But as web services became more and more sophisticated, more and more things were moved from their dedicated networks to the HTTP Protocol,
            3. Mail SMTP Simple mail transfer protocol
            4. VOIP
                a. Voice over ip
                b. Voice other the internet compared to over the telephone network.
    c. The World wide web
        i. The Internet
        ii. Web implementation
            1. URL (uniform resource locater)
                a. http://sdu.dk/information/odense.pdf
        iii. HTML(Hyper text markup language)
        iv. XML (eXtensible markup language)
        v. Client-side and server side Activities
                a. Client send request to server, and server serve user.
    d. Internet protocols
        i. The layered approach to Internet software

1. 
At each intermediate stop the network layer determines the direction in which the packet should be forwarded.
Prepares message and provides destination address
Application
Chops message into packets
Transport
Assigns intermediate address to each packet
Network
Transfers packet
Link
Network
Link
Network
Link
Application
Receives message
Transport
Collects packets and reassembles message
Network
Detects that packet has reached its final destination
Link
Receives packet
Origin
Intermediate stops
Final destination
2. Each application uses a port for transmitting data

ii. The TCP/IP protocol suite

1. Protocol package for handing data transmission over a network

e. Security

i. Forms of attack

1. DDOS and DOS
2. Malware
3. Spyware
4. Worm
5. Trojan horse
6. Phishing
7. Man in the middle attack
8. Social engineering attack

ii. Protection and cures

1. Firewall
2. Spam filter
3. Antivirus program

iii. Encryption

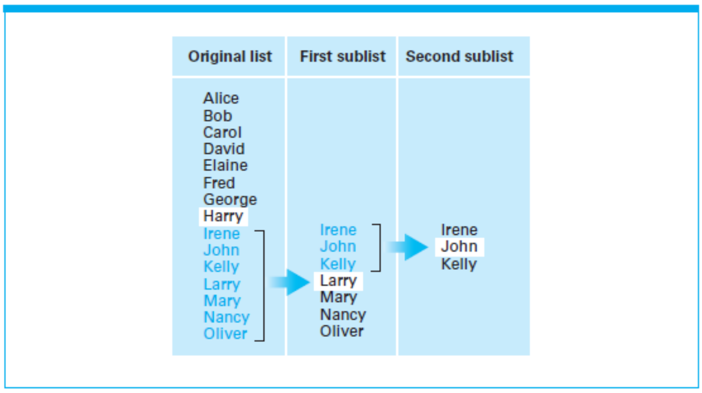
1. HTTPS
    a. HTTP (SSL)
2. SSL
    a. Secure socket layer

iv. Legal approaches to network security

1. Nope nope and nope

5. Algorithms
   a. The concept of an algorithm
      i. The formal definition of an algorithm
         1. “an Algorithm is an ordered set of unambiguous, executable steps that defines a terminating process.
      ii. The abstract nature of an algorithm
         1. Algorithms can be defined in many different ways but can do the same.
   b. Algorithm representation
      i. Primitives
         1. In simple English first do, then do etc
      ii. Pseudocode
         1. If, else, while, for statement and logic like that so it can be easily implemented in code.
   c. Algorithm discovery
      i. The art of problem solving
         1. Understand the problem
         2. Get an idea of how a algorithm may be able to solve the problem
         3. Formulate a algorithm and represent it as a problem
         4. Test and evaluate the result for accuracy and if it can be used in to real world.
   d. Iterative structures
      i. The Sequential search algorithm
         1. do something until statement is true/false
         2. up to n operations (worst case)
         3. 1 best case
      ii. Loop control
         1. While and for loops, gives the programmer more control and it makes the code more flexible
      iii. The insertion sort algorithm
         1. Worst case n^2
         2. Base case n
   e. Recursive structures
      i. The binary search algorithm
         1. Find something in a already sorted list.
         2.

Figure 5.12 Applying our strategy to search a list for the entry John



ii. Recursion control

1. Having a if statement before the recursive call to avoid a infinite function

f. Efficiency and correctness

i. You have to use the correct algorithms in the right applications to avoid overhead and to reduce run times for the problem.

6. Database systems
   a. Database fundamentals
      i. The significance of database systems
         1. A database is super optimized compared the privies solution where end part of the company had one file on one computer or a solution like it. When the database system was invented, it was possible to have the whole company in the same data set, which would mean that you would all have a unified and live set of the data, this meant that you could work most faster and spend less time checking records and unifying them.
      ii. The role of schemas
         1. The schema defines that database what information should be in what tables, and what type of data is allowed, it’s also used to set user rights in database systems
      iii. Database management systems
         1. System used to interact with the database itself, and the program the programs you designs interacts with and will maintain the integrity of the database.
      iv. Database models
         1. Relational database
         2. Object oriented database
   b. The relations model
         1. Tables called relations with rows called tuples with information
      ii. Issues of relation design
            a. You should only have one concept in a relation, as if you should not have employee Job and assignment in the same relation.
      iii. Relational operations
         1. Select, Project, and join.
      iv. SQL
         1. Select form job
         2. From assignment, job
         3. Where assignment.JobID = Job.JobID
         4. And assignment.termdata = “*”:
   c. Object-oriented databases
      i. In a object orientated database the data is linked to each other with from the privies selection.
   d. Maintaining database integrity
         1. In small private data bases a corrupt or loss of data are generally not a problem as then user can fast repair the data or load from a small back-up and bring the data up to data relatively fast.
         2. In a enterprise sitting where a multiuser database is in use a error can have disasters effect when a problem with the database arises and most systems will be down and a whole loss of income in imminent.
      ii. The commit rollback protocol locking
         1. Commit point is when the transaction has been recorded by the log in the database.

        2. Using the log a rollback can be performed to undo all the transactions until a point in time.
        3. But a rollback can result in a cascading rollback where other transactions also have be undone due to whats undone.

    iii. Locking

        1. Lock a set data so only one program can edit it at the time.

e. Traditional file structure

    i. Sequential files

        1. Things are stored in 1 long string.3

    ii. Indexed files

        1. Information is stored in blocks with a key that makes its faster to get information from inside the file

    iii. Hash files

        1. Works like the indexed file except the key points at the information needed.

f. Data mining

    i. Analyzing of large sets of data to get useable information,

g. Social impact of database technology

    i. This means information is faster and easier to find and get, therefor making it harder to hide information from the public.

- 7. Artificial intelligence
  - a. Intelligence and machines
    - i. Intelligent agent (IA)
      - 1. A “AI” that can solve a problem in a limited realm
        - a. Robot, self-driving car, autonomous plane or a character in a video game.
          - i. Responds to stimuli from its environment
    - ii. Research methodologies
      - 1. Engineering track
        - a. Working very well, getting a program to solve a very specific task
      - 2. Computational understanding
        - a. Making a general AI
    - iii. The Turing test
      - 1. Having a conversation with the AI without being able to tell the difference from a human and the computer for 5 minutes
  - b. Perception
    - i. Understanding images
      - 1. Easy to design a specific system to handle one puzzle but hard to design a program to handle all puzzles
      - 2. Its hard to make a program understand what things are in the image, what items etc.
    - ii. Language processing
      - 1. Making a program but understanding and being able to act upon what you’ve said to it.
  - c. Reasoning
    - i. Production systems
      - 1. A collection of states
        - a. All states of the puzzle and ranked what closer to being solved.
      - 2. A collection of productions
        - a. All the possible moves that the program to make to come closer to being solved
      - 3. A control system
        - a. A system doing the calculation about what moves to do. And a directed graph of what to do.
    - ii. Search trees
      - 1. A tree leading from the start point to the goal point in the shortest or most efficient way.
    - iii. Heuristics
      - 1. From goal to the starting point.
  - d. Additional areas of research
    - i. Representing and manipulating knowledge
      - 1. Just say something
    - ii. Learning
      - 1. Imitation
      - 2. Supervised training
      - 3. Reinforcement

a. Given a rule to rule for itself if it was a success or a failure

iii. Genetic algorithm

1. No clue atm

e. Artificial neural networks

i. Basis properties

1. Outputs 1 or 0 depending if the input exceeds the threshold value

ii. Training artificial neural networks

1. Compared to normal programming the values are not entered into the system but trained via supervised training.

iii. Associative memory

1. Can recognize patterns in the data at hand.

f. Robotics

i. Engeering AI stuff YAAY fun stuff, you know this shizz

g. Considering the consequences

i. Stamp example

1. We don’t know how an AI with react, and what it will do.

8. Theory of computation
   - a. Functions and their computation
      - i.
   - b. Turing machines
      - i. Turing machines fundamentals
      - ii. The church-turing thesis
   - c. Universal programing languages
      - i. The bare bones language
      - ii. Programming in bare bones
      - iii. The universality of bare bones
   - d. A noncomputable function
      - i. The halting problem
      - ii. The unsolvability of the halting problem
   - e. Complexity of problems
      - i. Measuring a problems complexity
      - ii. Polynomial versus nonpolynomial problems
      - iii. Np problems
   - f. Public-key Cryptography
      - i. Modular notation
      - ii. RSA public key cryptography