## RESOURCES

**PAST PAPERS - www.exam-mate.com**
**NOTES (1) - https://goo.gl/2AG5no**
**NOTES (2) - https://compsci2014.wikispaces.com**
**NOTES (3) - https://goo.gl/77PYY5**
**NOTES (4) - http://ibcomp.fis.edu/review/IBCS2018.html**
**NOTES (5) - http://www.emjbe.net/moodle/course/view.php?id=2**

**Some sections of this revision guide have been taken directly from sources listed above.**

## TABLE OF CONTENTS

# Topic 1: Systems Fundamentals

**1.1.1**

**Successful change should include these points**

- change will benefit the management and the aims of the company and monitor assumptions, risks, dependencies, costs, return on investment, dis-benefits and cultural issues affecting the aim of the company.
- Explain why the change is taking place, the benefits of successful change (what is in it for us, and you) as well as the details of the change (when? where? who is involved? how much will it cost? etc.)
- Devise an effective education, training system
- Counter resistance from the employees of companies and persuade them to change.
- Provide personal counseling (if required) to alleviate any change-related fears.
- Monitor the change and address any updates.

**Describe the need for change management**

The need for change management is based on the need to make sure that new systems work with the old systems.

**Definition - What does Legacy System mean?**

- A legacy system, in the context of computing, refers to outdated computer systems, programming languages or application software that are used instead of available upgraded versions.

**Disadvantages:**

- The cost of maintaining an older system can be higher than maintaining a more updated version. The staff might not be familiar with such an outdated version of the system.
- Tech support may no longer be available.
- The data in the current system might not be compatible with newer systems and conversion may not be possible.
- Legacy systems are usually more vulnerable to security threats due to lack of security patches.
- Integration of newer systems may be a complicated process as the two systems may have completely different technologies

**Software as a Service (SaaS)**

- A client typically accesses software on the internet via a web browser
    - hence clients needs internet connection to use the software.
    - Clients typically have to pay by a subscription either monthly or annually.
        - Costs of the Application Service Provider (ASP) will typically be higher than others in the industry as they must build and maintain the infrastructure needed to host the software.
- hosting system: it may become quite complicated to update the software.
    - Possible shutdown for clients or direct traffic to alternative servers

    - SaaS is generally updated far more regularly than software on a client's hardware, because:
- The application is hosted centrally, so it is the providers decision whether or not to update.
- The application only has a single configuration, making development testing faster.
- The application vendor has access to all customer data.
- The solution provider has access to user behavior within the application, making it easier to identify areas that need to be improved.

As the software is hosted on the internet it allows users to work together in real time.

**Criticism of Saas Privacy:**
Since you must send all your data to the ASP it allows them to access all your data, furthermore it makes the clients data more vulnerable to attack as hackers may hijack your connection.

**Alternative Installation Processes**
**Parallel running:**
You have a new system but you also run the old system while updating the features eventually getting rid of the old system.
**Pilot running:**
Taking a small group of people and testing the system with them before implementing it for the whole community.
**Direct changeover:**
Replace system without testing.
**Phased conversion:**
Implementing changes over time.
**Training session:**
Training the users of the new system beforehand with a limited time frame.

The installation process used in these examples are:
Printing at DIS- parallel running, pilot running, training session
Macs – Direct changeover, training session
BYOD – Direct changeover, phase, pilot, parallel, training session.

**What is Data Migration?**
Data Migration is the process of transferring data between storage types, format or computer systems.

**Problems that can arise**

- Data can become corrupted in transfer.
- Data is unusable whilst being transferred, for large files this can be problematic as well as for files that people are likely to be using.
- Cost is a problem that can arise in data migration because in order to migrate data to new storage you have to own the new storage. Physical space is also taken up by more storage.

**Problems with Data Migration**

These include incompatible file formats, data structures, validation rules, incomplete data transfer and international conventions on dates, currencies and character sets.

Disadvantages:

- Migration addresses the possible obsolescence of the data carrier, but does not address the fact that certain technologies which run the data may be abandoned altogether, leaving migration useless.
- Time consuming – migration is a continual process, which must be repeated every time a medium reaches obsolescence, for all data objects stored on a certain media.
- Costly - an institution must purchase additional data storage media at each migration." -Wikipedia

**Various Types of Testing**

Testing is very important in developing a computerised system, as it tries to ensure that the system works as expected. A system that does not work as expected (it is buggy) greatly reduces productivity and end user satisfaction. Testing is usually done in 2 stages. Testing in the first stage is often referred to as Alpha testing, while testing in the second stage is often referred to as Beta testing.

Alpha testing is done by the developers themselves, while Beta testing involves testing by real users with real data.

There are other types of testings such as:

- Debugging - It is done by the automatic program called debugger
- User acceptance testing - It is done by the final users
- White box testing -Tests internal structures or workings of the application
- Black box testing - Test the functionality of the application

**Importance of User Documentation**

User documentation is a crucial part of a system as it is the document that explains the working of the system to the user.

A well-made user documentation guides the user through using the system and thus increases productivity. If the user documentation is simple, system implementation can happen faster because users require less training to learn how to use the new system.

Users are non-technical people, they only need to know how to use the system. Therefore, the user documentation does not involve detailed explanations of how the system works.

A user documentation usually involves:

- Minimum hardware and software requirements

- Installation guide.
- How to start the system
- How to use different features of the system
- Screenshots explaining main features of the system
- Example inputs and outputs
- Explanations of error messages and troubleshooting guides
- Information to contact the developer of the system if an undocumented question arises

**Different Methods of Providing User Documentation**

Types of documentation:

1. Help files
2. Online support
3. Printed manuals

Help Files: supplied together with the system, called up with a system button

+ Accessible any time when using the program
+ Gives general instructions for system usage
+ Gives general instructions on solving major errors
- Can only be used AFTER system installation
- Deal with very general errors
- Lack of search capability

Printed manuals: printed on paper, supplied together with system

+ Can be read through by users before starting with new system
+ Always available
+ Gives help with system installation
- Can be lost
- Limited to small booklet with limited information about non-installation procedures
- May not be updated every time the system is updated

Online Support: special web service hosted by system's developer to provide documentation

- Often expensive compared to help files
- Useless without internet connection
- Live support does not work well if user is unfamiliar with computers ("I click THAT button and SOMETHING happened!")
+ Continuous revision by developer to deal with problems occurring most often (FAQs)
+ Provide option for live support, talking to a real human operator if problem arises which documentation has no answer to
+ Built-in search capabilities

**Methods of Delivering User Training**

- Training staff in using new system is very important as productivity greatly depends on how familiar users are with the system.
    - Good user training is an essential part of introducing a new system

**Methods of Delivering User Training:**

- Self-instruction
    - Users read manual or watch tutorial, or randomly do something in the system to figure out how it works.
    - Only suitable for experienced users
    - More comfortable as it gives freedom for making a study schedule
- Formal classes
    - Users sit in classroom, listen to instructor who shows and explains how to use the system
    - Useful for training large numbers of users as it is effective and cheap.
        - If too big, there is little time to deal with individual problems
- Remote/Online Training
    - Instructors trains a single user either by being in the same room or by some kind of remote connection (Skype, chat)
    - Most effective way of training as it can be suited to user's needs and abilities
    - Very expensive compared to other types of training

**Causes of Data Loss**

Two types of data loss causes:

1. **Malicious activities**
    a. Deleting data accidentally
    b. Not saving the data
    c. Hacking
    d. Viruses
    e. Data corruption
    f. Overheating
    g. Power failure
    h. Hardware damages
2. **Natural disasters**
    a. Floods
    b. Fires
    c. Storms
    d. Earthquakes

**Consequences of Data Loss in Specific Situation**

Loss in Schools

- Student information is lost
- School management becomes impossible without any data
- Reports, financial data, everything is lost

Loss in companies

-

- AMAG: pharmaceutical company in Boston with 300 employees.
- Their data loss problems began when an HR folder was moved within Google Drive, and didn't sync correctly. As a result, all files disappeared – including some that weren't even owned by the user moving the folder. The employee checked the trash bin, recycle bin, and desktop for a copy – but the data was gone.
- Luckily, Spanning Backup for Google Apps allowed AMAG to restore all files in just a matter of clicks, exactly as they were and where they were before.
- Without a backup and restore solution, the data is this important folder would have been lost forever, putting AMAG's compliance with regulations surrounding data availability and backup in jeopardy.

**Methods to Prevent Data Loss**

Failover systems, redundancy, removable media, offsite/online storage

Failover systems

- Data is stored in extra system which starts working once error is detected in main sys

Redundancy

- Same data is stored in different places
- RAID- Redundant Array of Independent Disks
    - Different levels, each with different methods of storing data
- Removable media
    - External hard drives, disks, USB
- Offsite Storage
    - Local storage that does not require internet
- Online Storage
    - Storing in clouds (requires internet)

**Strategies for managing releases and updates**

Update: improving, adding or removing certain things of a software
Upgrade: new version of the software
Patch: removing bugs and glitches

Advantages/Disadvantages of **Automatic Updates**

+ Easier for inexperienced users
+ No software problems because of user not updating or forgetting to update
+ Always done on time
- No choice for users to choose updates
- May not be desired by certain users
- May corrupt user files

**Advantages/Disadvantages of Manual Updates**

+ Gives users choice on whether or not to update
+ If update is badly made, it may prevent issues if users choose NOT to update
- Important updates may not be done on time
- Inexperienced users may forget to update which may lead to serious issues

**Managing Releases**

- Press release on developer website
- Notification within app or platform
- Give download option to update from developer/app/platform website

# Topic 1.2: System Design

1.2.1

- Hardware: anything within a computer you can physically touch
- Software: collection of programs and instructions that control the computer
- Peripheral: devices attached to computer that allow it to interact with its environment and exchange information
- Network: group of digital devices (computers) connected together in some way to share resources such as documents or files or printers
- Human resources: humans operating the computers

Types of peripherals
Input peripherals:

1. Keyboard
2. Mouse
3. Microphone
4. Webcam
5. Scanner

Output peripherals:

1. Screen
2. Speakers
3. Printer

**1.2.2 Roles of Computer in Networked World**

- Computer can be client, server or both at the same time

Client: computer accessing resources hosted by another computer, the server. Resources could be documents, printers, scanners, etc.
Server: computer hosting resources to be shared across the network, like documents, printers, etc.

Examples of Servers:

1. DNS servers: maps web address to IP addresses
2. Router: device that connects two different networks
3. Firewall: computer or software package which needs specific hardware, monitoring and protecting the network traffic of a computer or computer network.
    a. Decides which computers get access to the network based on a set of rules.
    b. Difference between firewall and antivirus is that firewall is used for the network.

4. Email server: maps email addresses to IP addresses

### 1.2.3 Social & Ethical Issues Associated with Networked World

- Loss of jobs due to computer automation
- Digital divide: more confidence with computer use have higher chances for success than those who don't.
- Distraction: limited face-to-face social interaction
- Security issues
- Privacy issues
- Reliability
- Copyright infringements
- Globalization

### 1.2.4 Relevant Stakeholders in Planning New System

Stakeholder: person or company with interest or concern in something. Stakeholders are:

- Developers
- End-users
- Critics
- Software manufacturers
- Brokers
- Investors
- Government

### 1.2.5 Methods to Obtaining Requirements from Stakeholders

**Observation ( ͡° ͜ʖ ͡°)**

Observation is the process of monitoring a system while it is at work.
Example: An awkward man stands behind a store clerk for one whole day, seeing how she works.

**Pros:**

- As it is not based on numerical statistics, it can explore topics in great detail and depth.
- Unobstructive observation will create extremely natural data.

**Cons:**

- A functioning system that is being observed will always act different than it does normally.
- The non-statistical data can be difficult to analyze and can cause generalizations.

**Surveys (2 types of surveys: online, face-to-face. 2 types of questions of surveys: restricted (yes/no questions), unrestricted)**

**Pros**

- Relatively easy to administer
- Can be developed in less time (compared to other data-collection methods)
- Can be administered remotely via online, mobile devices, mail, email, kiosk or telephone
- Conducted remotely can reduce or prevent geographical dependence
- Capable of collecting data from large number of respondents

- Numerous questions can be asked about a subject, giving extensive flexibility in data analysis
- With survey software, advanced statistical techniques can be utilized to analyze survey data to determine validity, reliability, and statistical significance, including the ability to analyze multiple variables.
- Broad range of data can be collected (e.g. attitude, opinion, value, belief, factual, etc)
- Standardized surveys are relatively free from several types of errors

**Cons**

- Respondents may not feel encouraged to provide accurate or honest answers
- Respondents may not be fully aware of their reasons for any given answer because of lack of memory on the subject, or even boredom.
- Number of respondents who choose to respond to a survey question may be different from those who chose not to respond, creating biased data.
- Survey question answer option could lead to unclear data because certain answer options may be interpreted differently by respondents. “The answer “somewhat agree” may represent different things to different subjects, and have its own meaning to each individual respondent. “Yes” or “no” answer options can also be problematic. Respondents may answer “no” if the option “only once” is not available.”

**Interviews (2 types: Structure (fixed questions) and unstructured)**

An interview is a face to face meeting, that consists of an interviewer asking the interviewee questions that should be pertinent to the topic of the interview.

**Pros**

- Direct feedback from stakeholders
- You can ask any question you want, and can change the direction of the interview
- People tend to be more truthful face to face.
- You can see how people physically reach to certain questions
- You can discuss a topic in depth

**Cons**

- It takes a long to interview everyone
- It is quite costly as people have to stop working to go to the interview and you have to pay the interviewers.
- Since it takes so long you are likely to choose a small group of people to interview and if you have badly selected this group then some of the information you collect may be useless.
- For an effective interview, interviewers must take a lot of time to prepare

The interviewees privacy be compromised if they are put in a position where they are asked about something that they do not want/need to answer.

**1.2.6 Appropriate techniques to gather information for a workable solution**

- **Examining current systems**
  - Looking at competing products
  - Organizational capabilities
  - Literature searches (web/book)

Observing the system at work can provide invaluable information as to how a system works and how it can be improved.

Most important things:

- Observe the users
- Workflow
- Inputs and outputs

**Techniques**

- Organizational capabilities: ability and capacity of an organization in terms of its human resources: number, quality, skills, and experience
- Examining the current system to get information about how it works and how it can be improved
- Looking at competing products to see what they include

### 1.2.7 Suitable Representations to Illustrate System Requirements

The ways to illustrate system requirements are system flowcharts, data flow diagrams, and structure charts.

**System flowcharts**: diagrams representing how different parts of the system are linked together and how the system should work in general.
**Data flow diagram**: diagram representing how data is moving through the system
**Structure Diagram**: diagrams representing the organization of a system, usually showing the different parts in hierarchical order.

### 1.2.8 Purpose of Prototypes to Demonstrate Proposed System to Client

**Prototype:** An original type, form, or instance serving as a basis or standard for later stages.

Benefits:

1. Software designer and implementer can get valuable feedback from users early in the project
2. Client and contractor can compare if software made matches the software specification according to which the program was built.
3. Allows engineer some insight into the accuracy of initial project estimates and whether the deadlines and milestones proposed can be successfully met.

**Process of Prototyping**:

1. **Identify basic requirements:** Determine basic requirements including the input and output information desired. Details, such as security, can typically be ignored.
2. **Develop Initial Prototype:** The initial prototype is developed that includes only user interfaces.
3. **Review:** The customers, including end-users, examine the prototype and provide feedback on additions or changes.
4. **Revision & Enhancement**: Using feedback, specifications and prototype can be improved. Negotiation of contract scope may be necessary. If changes are introduced then a repeat of steps #3 and #4 may be needed.

### 1.2.9 Importance of Iteration during Design Process

- Design and creation of system usually happens in 5 stages: planning, analysis, design, implementation, maintenance
- However, maintenance is often left out because it involves the use of all 5 steps, meaning that the circle is effectively begins from the start again.
- Iterative process happens during the entire product life cycle.

- Iterative system helps you find problems quickly because it makes it easy to try it again.

Once design is completed, problems are noted that many test users had with it. These problems would be quickly fixed through an iteration which should be tested to make sure the “fixes” went through and problems that had occurred are solved.

- New usability problems can also be found using this method after design changes.
- Iterative design is aimed to teach you through messing up repetitively until you get it right in a more successful final product.
- When applied early in the development stage, significant cost savings are possible.

Iterative testing enables an objective assessment of the project's status. Inconsistencies among requirements, designs, and implementations are detected early.

### 1.2.10 Possible Consequences of Failing to Involve End-User in Design Process

- User may be unsatisfied with system
- System may be unsuited for users problem, affecting productivity
- Off-the-shelf software ↔tailored software
- Software incompatibility

Failing to include the end user in the design process can lead to the end product not meeting their requirements or being appropriate from the intended use. Less feedback is gained so the system might have a bias towards the developers.

### 1.2.11 Social and Ethical Issues Associated with Introduction of New IT Systems

**Reliability & Integrity**

- Reliability: refers to the operation of hardware, the design of software, the accuracy of data or the correspondence of data with the real world.
- Data: may be unreliable if it has been entered incorrectly or if it becomes outdated.
- The reliability of machines, software and data determines our confidence in their value.

- Integrity refers to safeguarding the accuracy and completeness of stored data.
- Data lacks integrity when it has been changed accidentally or tampered with.
- Examples of data losing integrity are where information is duplicated in a relational database and only one copy is updated or where data entries have been maliciously altered.

**Security**

- Security: refers to the protection of hardware, software, machines and networks from unauthorized access.
- Security measures include restricted access to machines and networks for certain employees or to prevent access by hackers.
- The degree of security of information systems largely determines society's confidence in the information contained in the systems.

**Privacy & Anonymity**

- Privacy: ability of individuals and groups to determine for themselves when, how and to what extent information about themselves is shared with others.
- At its extreme, privacy becomes anonymity when, a person uses it to conceal his or her true identity in order to cyberbully someone else.
- Conversely, excessive privacy could also conceal the perpetrators of criminal, terrorist or computer hacking acts from law enforcement agencies.

**Intellectual Property**

- Includes ideas, discoveries, writings, software, presentations of data, etc.
- Copyright, trademarks and patents exist to protect intellectual property. However, the easy and accurate duplication methods made available through IT can undermine such protection.

**Authenticity**

- Means establishing a user's identity beyond reasonable doubt.
- Authenticating the user is crucial in many scenarios, particularly in business and legal matters. A simple example of authentication is a user login to a network. A more advanced example would be the use of encrypted digital signatures in a business transaction or the use of watermarking on digital photographs.

**Digital Divide and Equality of Access**

- The growth of the use of IT systems has led to disparities in the use of, and access to, information technologies.
- Disparities exist internationally, within countries between different socio-economic groups as well as relatively homogenous groups.
- This may lead to groups or individuals without access to IT being disadvantaged.

**Surveillance**

- Use of IT to monitor the actions of people.
- May be used to track, record and assess employees' performance.
- Can be used to support claims for promotion or to ensure that employees follow the organization's internet policy appropriately.

**Standards & Protocols**

- Technical rules and conventions that enable compatibility and therefore facilitate communication or interoperability between different IT systems and their components.
- They might govern the design and use of hardware, software and information. For example, the communication protocols used on the internet, the ASCII representations for characters, or the design of the printer port on a personal computer are all governed by standards.

### 1.2.12 Usability

Usability = Ergonomics + Accessibility

- **Usability**: the ease of use and learnability of a hardware or software. The object of use can be a software application, website, machine, process, or anything a human interacts with.
- **Accessibility**: Refers to the design of products for people with disabilities or specific needs.
- **Ergonomics**: The scientific discipline concerned with the understanding of interactions among humans and other elements of a system.
- **Learnability**: How easy is it for users to accomplish basic tasks the first time they encounter the design?
- **Efficiency:** Once users have learned the design, how quickly can they perform tasks?
- **Memorability:** When users return to the design after a period of not using it, how easily can they establish proficiency?
- **Errors:** How many errors do users make, how severe are these errors, and how easily can they recover from the errors?
- **Satisfaction:** How pleasant is it to use the design?

### 1.2.13 Identify a range of usability problems with commonly used digital devices

**PC Usability issues**

- If one is blind, screen cannot be viewed
- If one is disabled (hands), typing or mouse control is not possible
- Lack of technical knowledge could result in difficulty in device connection

**Cell Phone Usability Issues**

- Too complex OS: users have a hard time with navigation
- Bad eyesight: issues with looking at small screen
- Health issues: qwerty usage may be hard
- Hearing issues: low speaker volumes will be a problem
- Outdoors: sunlight will make screen impossible to view
- The phone MUST fit the hands in order to avoid slipping

**Digital Camera**

- Health issues can cause a variety of limitations to using digital cameras

**MP3 Player Usability**

- Screen, Buttons and controls are too small

### 1.2.14 Improving Accessibility of Systems

- **Touch screens** - for inexperienced users
- **Voice recognition** - for people who cannot see
- **Text-to-speech, speech-to-text** - transforms speech to a text and vice versa
- **Braille keyboards/printers** - for people who are blind
- **Sip and Puff** - assistive technology used to send signals to a device using air pressure by "sipping" (inhaling) or "puffing" (exhaling).
- **Simpler interface with few buttons** - for inexperienced users
- **Track balls -** A trackball is a computer cursor control device used in many notebook and laptop computers and is designed for people who have difficulties moving their hand.

### 1.2.15 Usability Problems that can occur in a System

**Automatic feedback system** - Is the feedback of good enough quality, is it clear enough, is there enough feedback and is it given quickly enough?

**Online Feedback System** - Are the people providing feedback of the right skill level and is their feedback valid?

**Online payroll**

**The inputs to a payroll system are:**

- Employee code (used to lookup the employee's other details, e.g. name, bank account, etc.)
- Hours worked
- Rate of pay (e.g. $25 per hour)

**The outputs from a payroll system are:**

- A printed payslip (given to the employee to show how his/her pay was calculated)
- A cheque, or an EFT payment directly into the employee's bank account

**Disadvantages**

- Internet Service Provider (ISP) downtime
- Depending on the complexity of the system, employees may have trouble learning how to work with it.
- Employees might not have a bank in order to be payed
- This system might be complex to set up

**Voice Recognition**

**Several problems may occur with such system:**

1. The dictionary of the voice recognition software may be limited and doesn't include certain words such as slang terms.
2. If the user has a speech impediment then the software may have trouble recognizing what they are saying.
3. People from different regions of the world and backgrounds will all different dialects as well as accents which the system may not be able to deal with.
4. The system may not be able to recognize different words if the user speaks very quickly or too slowly.
5. Users may not want to speak about important and sensitive information in a public space. As well as people speaking in a lower volume when in public, the system may not be able to 'hear' the user.
6. Depending on the security capabilities of the voice recognition system.
7. If the user is in an area with a lot of ambient noise then the system may not be able to cancel out this noise.
8. If the word database is not stored locally and there are network errors then the system is not able to work.

### 1.2.16 Implications of Human-Machine Interaction

- **Moral**: Robot use in military. Is it okay?
- **Ethical**: Robots replacing humans in factories
- **Social**: less human interaction because of phones
- **Economic**: pertaining to the production and distribution
- **Environmental**: impact of human activity on nature in the name of technology

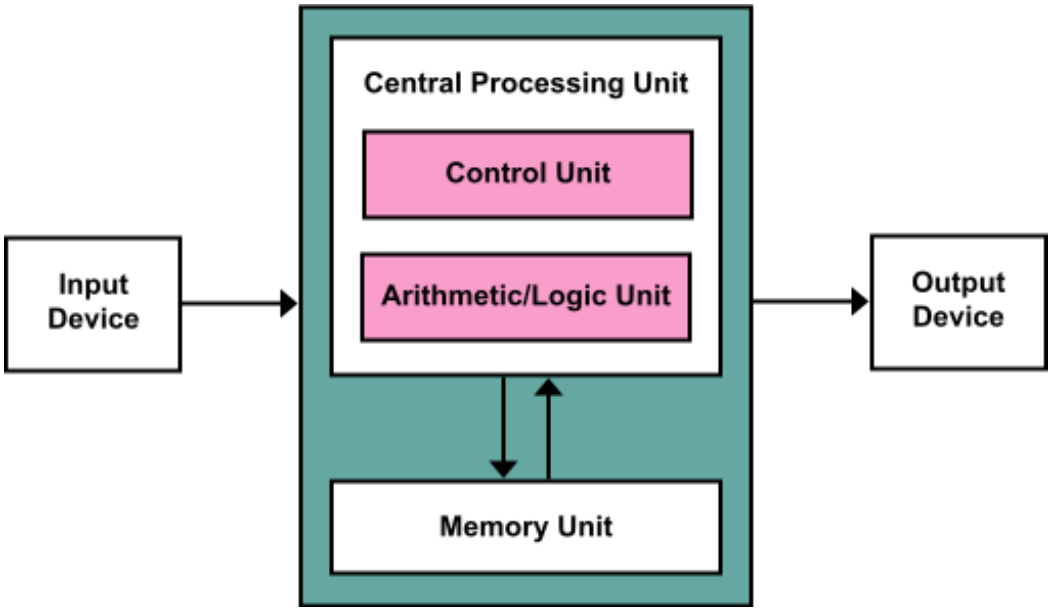
## Topic 2 & 6: Computer Organization | Resource Management

https://en.wikibooks.org/wiki/IB/Group_4/Computer_Science/Computer_Organisation

- 1944-45: data and instructions to manipulate data were logically the same and could be stored in the same place.
- Von Neumann Architecture: basis for computers today.

**von Neumann Architecture**

- Units that process information are separate from units that store information.
    - Leads to:




- **Memory Unit:** holds both data and instructions.
- **Arithmetic Logic Unit (ALU):** performs arithmetic and logic operations on data.
- **Input Unit**: Moves data from the outside world into the computer.
- **Output Unit:** Moves results from inside the computer to the outside world.
- **Control Unit (CU):** Acts as the stage unit to ensure that all the other components act in concert.

**Memory**

- Collection of cells, each with a unique physical address.
- Called cells instead of bytes because byte-capacity varies by machine.

**Addressability:** The number of bits stored in each addressable location in memory. Usually 8 bits.

**8 bits = 256 cells of memory (00000000)**

**Address: 00000000, but contents can be anything (like 11001001, 8! possibilities)**

Read right-to-left, beginning from zero.

**Arithmetic/Logic Unit**

- Capable of performing basic arithmetic operations such as adding, subtracting, multiplying, and dividing two numbers.
- Also capable of performing logical operations such as AND, OR, and NOT.
- Operates on words, a natural unit of data associated with a particular computer design.
- The processor can work on words (of 16 bits), double words (32 bits), and quadwords (64 bits).

- Most modern ALUs have a small number of special storage units called Registers.

**Registers: small storage area in the CPU used to store intermediate values or special data.**

- Contain one word.
- Used to store information that is needed again immediately.
- Access to registers is much faster than access to memory locations.

**Input/Output Units**

- Input Unit is a device through which data and programs from the outside world are entered into the computer. E.g. mice, cameras, scanners, keyboards.
- Output Unit is a device through which results stored in the computer memory are made available to the outside world. E.g. include printers and screen monitors.

**Control Unit**

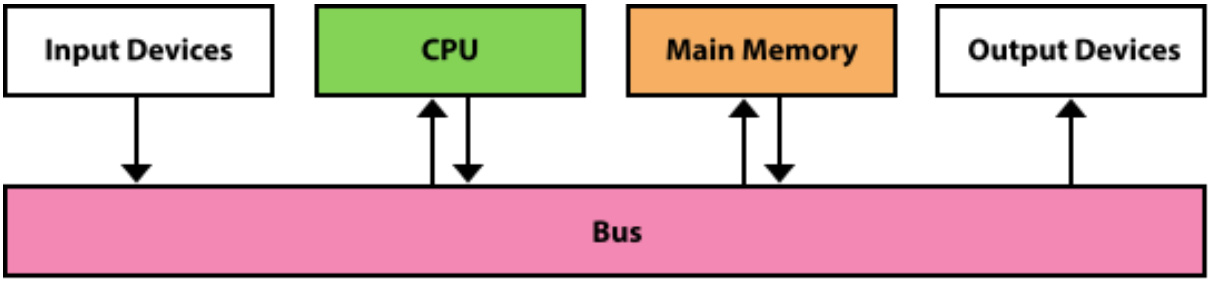
- Organizing force in the computer, in-charge of Fetch-Execute Cycle.

Two Registers in CU:

- Instruction Register: contains the instruction that is being executed
- Program Counter (PC) contains the address of the next instruction to be executed.

**Because the ALU and the control unit work so closely together they are often thought of as one unit called the Central Processing Unit (CPU).**

**Flow of Information in von Neumann Machines**




**Bus Width:** The number of bits that can be transferred in parallel over the bus
**Cache Memory:** type of small, high-speed memory used to hold frequently used data.

**Pipelining: technique used to break an instruction down into small overlapping steps.**

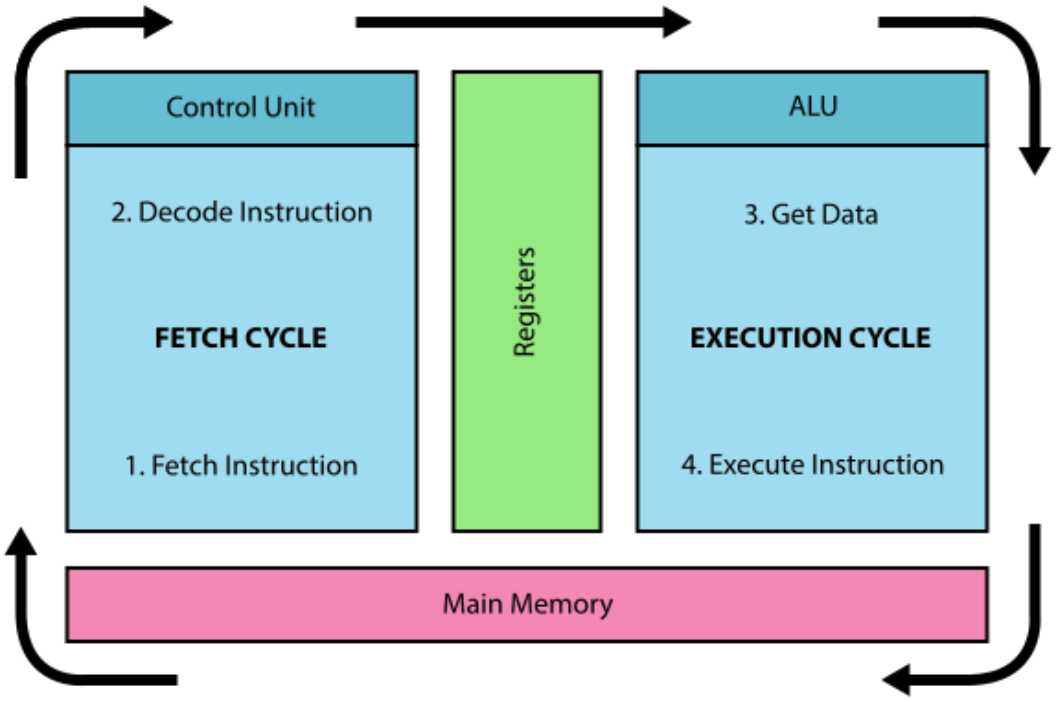
- **Used to speed up Fetch-Execute Cycle**

**Motherboard: main circuit of a computer**

***n-bit processors***

- n refers to number of bits in CPU general registers.
- Two n-bit numbers can be added with a single instruction.
- It also can refer to the width of the bus, which is the size of the addressable memory—but not always.
- *n* can refer to the width of the data bus—but not always.

**Fetch-Execute Cycle**




**Fetch the next Instruction:**

- PrgCntr increments one by one to point to the next instruction to be executed
- CU goes to the address in the memory address register which holds the address of the next instruction
- Takes it to the main memory through the address bus and returns it to the memory buffer (MBR) register via the data bus.
- MBR is a two way register that temporarily holds data fetched from the Memory (cache or RAM), makes a copy of the contents, and places the copy in the IR.
- If instruction needs additional data from memory, ALU sends an address to the memory bus, and the memory responds by returning the value at that location.

**Decode the Instruction:**

- The instruction is decoded into control signals.
- The logic of the circuitry in the CPU determines which operation is to be executed.
- Shows how instructions are built into circuitry.

**Get Data Needed:**

- The instruction to be executed may potentially require additional memory accesses to complete its task. If so, CU sends memory location, MBR returns value.

**Execute the Instruction:**

- Involves sending signals to the arithmetic/logic unit to carry out the processing. In the case of adding a number to a register, the operand is sent to the ALU and added to the contents of the register.

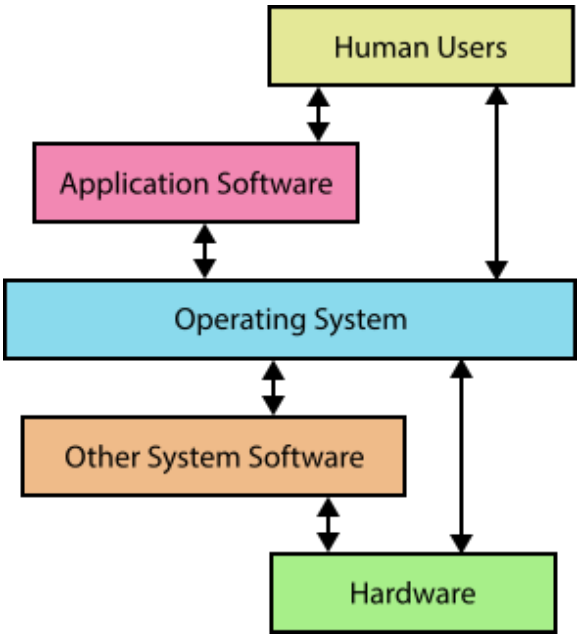
**RAM and ROM**

- **RAM - Random Access Memory**
  - Each cell (usually a byte) can be directly accessed
  - Inherent in the idea of being able to access each location is the ability to *change* the contents of each location.
  - Storing something else into that place can change the bit pattern in each cell.

- **ROM - Read-Only Memory**
  - Contents in locations cannot be changed. Their contents are permanent and cannot be altered by a second operation.
  - Placing the bit pattern in ROM is called *burning*. The bit pattern is burnt either at the time the ROM is manufactured or at the time the computer parts are assembled.
- **DIFFERENCES:**
  - RAM is volatile, ROM is not. RAM does not retain configuration after turn off.
    - Why?
      a. ROM is stable and cannot be changed.
      b. Used to store the instructions that the computer needs to start itself.
      c. Frequently used software is also stored in ROM so that the system does not have to read the software each time the machine is turned on.
      d. Main memory usually contains some ROM along with the general-purpose RAM.

**Secondary Storage Devices** - Magnetic Tapes, External Hard Drives, USB, CD, DVD
- Main memory is volatile and limited.
- Hence, storage devices where programs and data no longer being used is required.
- These types of devices are called *secondary* or *auxiliary* storage devices. They are input and output devices themselves.

**OS and its Applications**
- Two types: Application software, System software
- **Application software:** software designed to suit specific needs, like word processing, etc.
- **System Software:**
  - Manages a computer system at a more fundamental level.
  - Provides tools and environment creating & running application software.
  - Interacts directly with the hardware, provides more functionality than the hardware.
- **Operating System:**
  - Core of system software.
  - Manages computer resources, such as memory, and input/output devices
  - Provides an interface through which a human can interact with the computer.
  - The operating system allows an application program to interact with these other system resources.




**Memory, Process, CPU Management**

- **Multiprogramming:** Technique of keeping multiple programs in main memory at the same time, competing for the CPU.
- **Memory Management:** The act of keeping track of how and where programs are loaded in main memory.
- **Process Management:** The act of keeping track of information for active processes.
- **CPU Scheduling:** The act of determining which process in memory is given access to the CPU so that it may execute.

**PROCESS MANAGEMENT:**

1. First Come First Serve - Non Preemptive - cannot be interrupted
2. Priority Scheduling - Non Preemptive - cannot be interrupted
3. Shortest Job First - Non Preemptive - cannot be interrupted
4. Round Robin - Preemptive - can be interrupted

**MEMORY MANAGEMENT**

- **Single Contiguous Memory Management** - one program and OS in memory.
- **Partition Memory Management**
    - Memory is broken up into different parts. Has the operating system and any number of programs running at the same time through these different partitions. The partitions have a base and a bound register.

    - **Base register:** A register that holds the beginning address of the current partition (the one that is running)
    - **Bounds register:** A register that holds the length of the current partition

TABLE OF CONTENTS

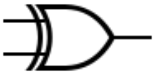


**Types of Partition:**

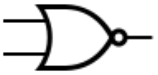
1. **Partition selection First fit:** Allocate program to the first partition big enough to hold it
2. **Best fit:** Allocated program to the smallest partition big enough to hold it
3. **Worst fit:** Allocation of program to the largest partition big enough to hold it

- **Paged Memory Management**
  - A technique in which processes are divided into fixed-size pages and stored in memory frames when loaded. Frame
    - **Frame:** A fixed-size portion of main memory that holds a process page
    - **Page:** A fixed-size portion of a process that is stored into a memory frame

**Logic Gates**

1. **NOT**
   a. X = A' || B
   b. A  X  if A = 0, X = 1, A = 1, X = 0
2. **AND**
   a. X = A • B
   b. A, B  X  if A & B = 1, then X = 1, else 0
3. **OR**
   a. X = A + B
   b. A, B  X  if A & B = 0, then X = 0, else 1
4. **XOR**
   a. X = A ⊕ B
   b. A, B  X  if A & B = same input, X = 0, else 1
5. **NAND**
   a. X = (A • B)'
   b. Opposite of AND, AND with inverter
   c. A, B  X  values opposite of AND
6. **NOR**
   a. X = (A + B)'

b. Opposite of OR, OR with inverter

c. A, B  X values opposite of OR

# Topic 3: Networks

- A computer network is a collection of computing devices that are connected in various ways to communicate and share resources.
- Used to share both tangible and intangible sources.

- Computer networks contain devices other than computers. Printers, for instance, can be connected directly to a network so that anyone on the network can print to them.
- Networks also contain a variety of devices for handing network traffic. We use the generic term node or host to refer to any device on a network.

- Key issue related to computer networks: bandwidth or data transfer rate.
- Bandwidth: speed at which data is moved from one place to another on a network.

### 3.1.1 Identify different types of Networks

**Types of Networks:**

1. **LAN - Local Area Network**
    a. Connects network devices over a relatively short distance.
    b. Examples include office buildings or schools.
    c. A home may have a LAN, but can include multiple LANs within it.
    d. In TCP/IP networking, LAN is qualsiasi used as a single IP subnet and tend to use a physical connection between nodes, such as that via Ethernet cables.
2. **VLAN - Virtual Local Area Network**
    a. Network of computers that behave as if they are connected to the same topology, but are actually physically located in different segments of a LAN.
    b. Configurations are generally comprised and maintained via software rather than hardware.
    c. Makes the network extremely flexible:
        i. Allows computers to be removed from the VLAN and located elsewhere without having to physically rewire the whole system to fit the new configuration.
3. **WAN - Wide Area Network**
    a. Geographically-dispersed group of two or more LANs.
    b. Permits communication among smaller networks.
    c. Uses one node set up as a gateway to handle all communication going through that LAN and other networks.
4. **SAN - Storage Area Network**

   a. Type of local area network (LAN) designed to handle large data transfers.
   b. Typically supports data storage, retrieval and replication on business networks using high-end servers, multiple disk arrays and Fibre Channel interconnection technology.

5. **WLAN - Wireless Local Area Network**
   a. Provides wireless network communication over short distances using radio or infrared signals instead of traditional network cabling.
   b. Typically extends an existing wired local area network.
   c. Built by attaching a device called the access point to the edge of the wired network.
   d. Clients communicate with the access point using a wireless network adapter similar in function to a traditional Ethernet adapter.
6. **VPN - Virtual Private Network**
   a. Type of WAN supplying network connectivity over a possibly long physical connection.
   b. Key feature of a VPN: ability to use public networks like the Internet rather than rely on private leased lines using a method called tunnelling, a VPN use the same hardware infrastructure as existing Internet or intranet links.
   c. Includes various security mechanisms to protect the virtual, private connections.
   d. Implement restricted-access networks that utilise the same cabling and routers as a public network, and they do so without sacrificing features or basic security.
7. **Client/Server**

- One or more computers act as a server and the other computers on the network request services from the server.
- A server, sometimes called a host computer, controls access to the hardware, software and other resources on the network and provides a centralized storage area for programs, data, and information.
- The clients are other computers and mobile devices on the network that rely on the server for its resources.

8. **P2P - Peer-to-Peer Network**
   a. Approach to computer networking where all computers share equivalent responsibility for processing data.
   b. Differs from client-server networking, where certain devices have responsibility for providing or serving data and other devices consume or otherwise act as clients of those servers.
   **c. Key features:**
      **i.** Two or more computers share equivalent responsibility for processing data
      **ii.** Do not have a central server, data is stored on all peer devices and shared among them as needed
   **d. Advantages:**
      i. Easy setup: does not require large infrastructure
      ii. Cost effective: less infrastructure and specialized equipment means even the simplest devices can be part of a P2P

iii. There is no need for a network administrator, as since each device has its share of "server responsibility", the user of each device is the network administrator
iv. Failure of one node seldom affects the entire network, as other nodes can pick up the load

e. **Disadvantages:**
i. Data recovery is difficult as data is stored on each individual node rather than a central server
ii. Poor security - users rarely need to login to their workstations, and the maximum security present are permissions, which are easy to bypass
iii. User performance can be affected if every other node is trying to access data from a particular node.
iv. Variable filing system may reduce efficiency and cause difficulty in data accessing as data storage is ultimately up to the device user/administrator

9. **Network Topologies**: layout of computers and devices in a communications network
a. bus network consists of single central cable which all computers and other devices connect
i. bus is the physical cable; transmits data, instructions, and information
ii. popular on LAN b/c inexpensive, easy to install; computers and other devices can be attached and detached at any point
iii. failure of one device does not affect the rest of the network
iv. biggest risk is if the bus becomes inoperable
b. ring network: cable forms a closed loop with all computers/devices arranged along the ring
i. data transmitted travels from device to device around the entire ring in one direction
ii. can span a larger distance but more difficult to install (usually for LANs)
iii. stuff after a failed device cannot function
c. star network: all computers/devices connect to a central device
i. central connection point is the hub/switch
ii. if one node fails, others are not connected
iii. if the hub/switch fails, everything goes down (large star networks keep backup hubs/switches

**3.1.2 & 3.1.3**

- Early in the development of networks, commercial vendors came out with a variety of technologies that they hoped businesses would adopt.
- Problem? proprietary systems were developed with their own particular nuances and did not permit communication between networks of differing types.
- As network technologies grew, the need for interoperability became clear; we needed a way of computing systems made by different vendors to communicate.

- **ASCII** - Before networks computers were all non-standard, custom built devices. They used "custom" storage methods, including non-standard character-code

systems. Once there were enough computers containing enough data to make communication and data exchange useful, standard storage systems were needed. Then ASCII was invented in the early 1960's, making it possible to exchange data and information easily. Original- 128 bits (7-bit binary + 1 parity code) - later 256 codes without parity bit and 8-bit binary.

- **UNICODE -** ASCII did not include codes for "foreign" languages like Chinese, Russian, Persian, etc. UNICODE is 16-bits, providing codes for 65536 different characters.
- **Ethernet -** a standard for LAN connections. This includes a communication protocol, as well as a standard type of connection card and standard cables. Without such standards, we could not ensure that new machines could connect successfully to an existing network.

**Open System:** system based on a common model of network architecture and a suite of protocols used in its implementation. Maximise the opportunity for interoperability.

**Open Systems Interconnection Reference Model (OSI Model):**

| **OSI Model** | | | |
|---|---|---|---|
| **Host Layers** | **Data** | Application | Network process to application |
| | | Presentation | convert machine dependent data to machine independent data |
| | | Session | managing sessions between applications |
| | **Segment** | Transport | Reliable delivery of packets |
| **Media Layers** | **Packet** | Network | Addressing, routing and delivery of datagrams |
| | **Bit / Frame** | Data link | reliable direct point-to-point data connection |
| | **Bit** | Physical | direct point-to-point data connection. (not reliable) |

- Each layer deals with a particular aspect of network communication.
- The highest level deals with issues that relate most specifically to the application program in question.
- The lowest layer deals with the most basic electrical and mechanical issues of the physical transmission medium.

**IMPORTANT: today's networking technology is only possible through OSI Model.**

**3.1.4 & 3.1.5**

- VPNs require a protocol that emphasises authentication and encryption. Authentication allows VPN clients and servers to correctly establish the identity of people on the network and encryption allows potentially sensitive data to be hidden.
- Several protocols like SOCKS and IPsec have become popular due to VPN development.

- A VPN is one solution to establishing long-distance and/or secured network connections.
- Normally deployed by businesses or organisations rather than by individuals.

A VPN can save an organisation money in several situations:

- Eliminating the need for expensive long-distance leased lines.
- Reducing long-distance telephone charges.
- Offloading support costs.

However, limitations exist.

- VPNs require detailed understanding of network security issues and careful installation / configuration to ensure sufficient protection on a public network
- The reliability and performance of an Internet-based VPN is not under an organisation's direct control.
- Historically, VPN products and solutions from different vendors have not always been compatible due to issues with VPN technology standards. Can cause technical failures.

**3.1.6 Data Transmission - Define protocol & data packet**

**Protocol:**

- rules/procedures for transmitting data between devices
- defines structure and how send/received
    - Examples:
        - **TCP/IP = Transmission Control Protocol / Internet Protocol**
        - includes IP addresses, packet-switching strategies, error correction strategies, handshaking methodologies, etc
    - **HTTP = HyperText Transport Protocol**
        - used by clients and servers in the WWW
    - **FTP - File Transfer Protocol**

**Data Packet:**

- unit of binary data, that can be routed through a network
- packet formats
    - header = destination, metadata
    - body
    - sometimes footer — parity checking

- also called datagram

**Definition**:

- Protocols are needed to ensure that the communication between the different networks functions. Networks can be structured in a different way, but structuring the packets send in the same way is important to ensure that the data can be received and interpreted.
- used to standardise transmission and ensure connectivity

**How do protocols help?**
Protocols are a set of RULES that control how communication happens. For example, if a client sends a request to a server, the protocol might require an immediate response - either the requested data or a message saying that the server is busy and there may be a delay. Even if the data will be sent quickly, the server might still send an **acknowledge** message, saying that it received the request. Another example is **handshaking**. When two devices start communicating, they must agree on the **transmission speed** that will be used. If the speed is set too high, there may be lots and lots of errors, resulting in lots and lots of retransmissions, meaning the high transmission speed actually slows down communication. Packet sizes, error-detection protocols and flow control (choose a different channel or server) all contribute to fewer problems and overall faster data transfer.

**3.1.7 Explain why protocols are necessary**

- **Integrity -** means that the data arrives unchanged, reliably. It is common that electromagnetic interference or other faults **corrupt** data during transmission.
- **Error Checking -** The data is sent in **packets** and each packet contains some error-checking codes, for example a **checksum** for the entire packet, which is a complicated calculation done on all the bytes in the packet. When a packet arrives, the recipient performs the checksum algorithm to check whether the result matches the result that was transmitted. If the results don't match, the recipient sends an error message back to the transmitter and asks for a **retransmission.**
- **Parity Check -** For smaller transmissions, such as a password in a LAN, we might use **parity checks.** This is a single **bit** that accompanies a **Byte**, tellings whether the sum of all the bits is **even** or **odd.** Again, the recipient checks whether the received parity bit correctly matches the received Byte - if not it asks for retransmission
- **Flow Control -** If a server is busy, like CNN during an important news event, then a client might request a web-page and the web-page does not arrive quickly. Then the client might assume that the request never arrived, so it sends another request. If lots of clients are making repeated requests, the server will **queue-up** all the requests, making a longer and longer list of unfinished work. This can make the server slower and slower, and eventually the server needs to be shut down. **Flow Control** should prevent such disasters, by cleverly managing the flow of data and requests. For example, it's better to immediately send back a "busy" message, rather than just NOT sending the requested data - causing the client to wait an unpredictable amount of time. If multiple servers are available, perhaps in various

geographical locations (like Google's servers) flow control can reroute requests to a less busy server.

- **Deadlock -** when two (or more) competing tasks are each preventing each other from finishing. So the system keeps trying over and over again to finish one of the tasks, but keeps getting interrupted by the other. In modern OS's, this seldom happens, but a common example is a PC that has too many processes running and is unsuccessfully trying to "swap" some of them to virtual memory. Every swap operation takes time, and this overhead makes the system unstable - we call this "thrashing". Deadlock implies that the problem is NEVER going to end.
- **Congestion -** similar to deadlock, but only causing a significant slowdown. This usually refers to the communication channels (wires and routers) rather than the servers.
- interoperability — to ensure different network hardwares work
- connects products from different companies
- bridges gap between software and hardware
- ensures different versions can be supported

### 3.1.8 Explain why data transmission speeds can vary

- The packets of a message may take different routes on their way to the final destination.
- Therefore, they may arrive in a different order than the way they were sent. The packets must be put into the proper order once again, and then combined to form the original message.
- A packet may make several intermediate hops between computers on various networks before it reaches its final destination.
- Network devices called routers direct the packets as they move between networks. Intermediate routers don't plan out the packet's entire course.
- Eventually a message reaches a router that knows where the destination machine is. If a path is blocked due to a down machine, or if a path currently has a lot of network traffic, a router might send a packet along an alternative route.
- If a communication line spans a long distance, such as across an ocean, devices called repeaters are installed periodically along the line to strengthen and propagate the signal.

### 3.1.9 Explain why data may need to be compressed before transmission

**Compression** is commonly used to reduce file sizes and hence reduce the overall amount of data being transmitted (traffic). This is effective as long as the client receiving the data is powerful enough to decompress the data quickly. If not, then we can use **lossy compression**, where some details are simply removed (lost). A typical example is to send a low-resolution version of a video, say 320x200, rather than an HD (high definition) version at 1920x1080.

### 3.1.10 Explain transmission media

**Metal cables (copper) -** CAT 5 cables work reliably up to 100 meters, at speeds like 100 Megabits per second. Cables can be relatively expensive in a large building with a long cable to each client, and with substantial installation costs for drilling holes and making connections. But the cables are highly reliable and provide relatively good security, since physical access is required if someone wishes to steal data.
**Wireless (WIFI)** - radio signals like 802.11n, which supports well over 100 Megabits per second. Unfortunately WIFI is subject to substantial interference from steel in walls and other radio signals. WIFI is generally inexpensive to install, especially because many devices can share the same "access point" (hub), without needing a cable for each device. The range is substantially less than 100 meters (more like 30 m). Reliability is generally good due to adjustable transmission speeds and good error-detection, but security is poor since the signal can be received outside the building.
**Fibre Optic -** tiny glass fibres, bundled inside a protective cable wall. The fibres carry light (optical signals, rather than electricity. The light signals are considerably faster than electrical signals, they are NOT disrupted by electromagnetic interference, and the optical fibres are a lot thinner than metal cables - so one optical fibre cable carries a lot more data (higher bandwidth) than similarly sized copper cable. Hence speed (actually bandwidth) is much higher. But fibre cables are a lot more expensive than copper, requiring significantly more sophisticated manufacturing techniques. Also, the fibre must connect into a special box that converts the optical signals to electronic signals, and these boxes are also expensive. Besides high bandwidth, one large attraction of fibre is that it will not attract lightning strikes and will not "short-out" when wet, making it very popular for use outside of buildings. The light signals decay slowly, so distances of 50 kilometers or more are possible, making fibre optic the principal medium for the Internet Backbone. "Hacking" into a fibre optic signal is considerably more difficult than tapping into a copper cable, so security is considerably better.

### 3.1.10 Packet Switching

**Packet switching:** a mode of data transmission in which a message is broken into a number of parts that are sent independently, over whatever route is optimum for each packet, and reassembled at the destination.

- **File assigned header, identification and sequence code**
- **File broken down into pieces**
- **File released onto transmission media through routers/modems**
- **File pieces bounce off routers and switches, use Dijkstra's algorithm to identify another router nearest to destination**
- **File pieces arrive at destination**
- **Destination router sorts and reconstructs pieces as per sequence code**
- **Destination router reroutes reconstructed files to connected device(s).**

### 3.1.12 Advantages/ Disadvantages of Wireless Networks

**Wireless Networking**

- Advantages of wireless networks include the flexibility of their ad-hoc situation when additional workstation was required
- The implementation cost is cheaper than wired network
- Ideal for the non-reachable places such as across river or mountain or rural area
- Ideal for temporary network setups.

- There is a reduction in speed compared to wired networks when signal passes through permeable air.
- Less secure because cracker's laptop can act as access point. If you connected to their laptop, they'll read all your information sensitive information.
- Greatly affected by their surrounding
- Signal strength affected by objects blocking its path, interference, and attenuation.

**Characteristics of Wireless Networks**

**WiFi** - A Wireless Access Point (WAP) will provide a wireless signal that allows data transfer between it and a device. The current protocol for a WAP is 802.11, which is able to communicate with approx. 30 clients over a 100m radius and has a theoretical top speed of 300 Mbps. Will normally have WPA2 encryption, which is considered secure along with a password.

**WiMAX** - Worldwide Interoperability for Microwave Access (WiMAX) "is a wireless communications standard designed to provide 30 to 40 megabit-per-second data rates, with the 2011 update providing up to 1 Gbit/s for fixed stations."

- a standards-based technology enabling the delivery of last mile wireless broadband access as an alternative to cable and DSL
- WiMAX is a part of fourth generation wireless-communication technology (4G) and is able to cover a staggering radius of about 50 km.

**3.1.13 Hardware/Software Components of Networks**

**WIFI Hardware**

- **Access points** - radio receivers that communicate with digital devices like laptops and Smartphones.
- **Cables -** Most access points are then connected by a **cable** to a telephone line or into a cabled LAN.
- **WIFI Enabled Device** - computer, Smartphone, television, etc. Any device with an antenna and a WIFI chip/card and a suitable OS. If a device does NOT contain WIFI support, a "dongle" might be added and plugged into a USB port, assuming the OS will support this. This might not work in an older computer.
- **Mobile Devices -** devices like tablets and Smartphones are purposely designed for portability, depending on batteries and designed to be small and easily carried

around. These are virtually all intended for WIFI connections, and probably don't contain a network cable connector (RJ45).

**WIFI Software**

WIFI support is provided as standard **modules** in most OS's installed in devices requiring the support. Application software is normally the same for WIFI or cabled networks, so no specific applications are required. It might be the case that NOT ALL browsers run on a device - say a Smartphone - but probably SOME browser will work.

- **Cloud -** Mobile devices make extensive use of "cloud computing", especially online data storage and browser-based "apps". This adds to the portability of the device, making personal data available around the world by storing it on a web-server.

**Different Methods of Security**

1. Encryption

- **Wireless Encryption**
  - **Wired Equivalent Privacy (WEP)**
    - Often mistakenly referred to as the “Wireless Encryption Protocol”
    - Depreciated method of wireless security which can be cracked by a malicious individual within minutes (PDF link).
    - Usually easily identifiable, since the password is generally a series of hexadecimal characters (that is: 0–9, A–F).
  - **Wi-Fi Protected Access (WPA)**
    - WPA, and it’s more secure and more resource intensive successor
    - WPA2 use a few different methods to encrypt traffic.
    - The differences between WPA and WPA2 lie in the encryption methods used, TKIP and the AES-based CCMP respectively.
    - There are flaws in the older TKIP protocols that can also be used to gain access to your network, but it tends to be more compatible with older hardware
  - **Pre-shared Key (PSK)**
    - PSK is the most common method of securing a wireless network
    - Easiest to implement and doesn't require a separate authentication server
    - Everyone on your secure wireless network connects to access points utilizing the same password, which is probably known by everyone.
    - The trouble with PSK is a new key needs to be sanctioned if restrictions need to be made.
  - **User ID**
    - Also known as a username, name a user logs in to a system with. Usually used alongside a password it is one half of a set of credentials a user knows in order to access a system.
  - **MAC Address**

- A **media access control address** (**MAC address**) of a device is a unique identifier assigned to network interfaces for communications at the data link layer of a network segment. Logically, MAC addresses are used in the media access control protocol sublayer of the OSI reference model.

- **Authentication**
  - One factor authentication:
    - Password only (user “KNOWS”)
  - Two factor authentication:
    - Password and auto-generated code (user “HAS”)
  - Three factor authentication:
    - Retina scan or fingerprint (user “IS”)
- **Firewall**
  - Primary objective is to control the incoming and outgoing network traffic by analyzing the data packets and determining whether it should be allowed through or not, based on a predetermined rule set.
- **Physical Security**
  - Something that is physically in the way of someone breaking into a system. E.g. a door, or walls, or security guards

**Advantages & Disadvantages of Firewall, MAC Address & Authentication:**

1. **Firewalls**
   a. **Advantages**
      - Relatively inexpensive or free for personal use.
      - New releases are becoming user friendly.
      - Some firewalls but not all can detect viruses, worms, Trojan horses, or data collectors.
      - A firewall blocks evil packets from being permitted to reach a place where they can do harm.

   b. **Disadvantages**
      - Firewalls evolve due to cracker's ability to circumvent the increases.
      - Firewalls cannot protect you from internal sabotage within a network or from allowing other user’s access to your PC.
      - Firewalls offer weak defense from viruses.
2. **MAC Address**
   a. **Advantages**
      - There is no attachment cost to devices that connect to the network. The policy is set on a router or switch, and the equipment attached either is permitted or it is not.

**b. Disadvantages**

- The disadvantage to MAC filtering is that it is easy to spoof. Advisory can sit on the wire and just listen to traffic to and from permitted MAC addresses. Then, the advisory can change his MAC address to a permitted one, and in most cases obtain access to the network.

# Topic 4: Computational Thinking & Programming

**Topic 4—Computational thinking, problem-solving and program *Notes by Dave Mulkey* (45 hours) *15, Germany***

**4.1 General principles (10 hours)**

This should not be taught as a separate topic but must be incorporated and connected to all sections—especially flowcharts, pseudocode and programming in the SL/HL core and abstract data structures (HL extension). It is essential that these elements are not addressed in isolation—they have to be approached as a whole.
The basic ideas and their application should be illustrated with non-computer examples. Each basic idea should then be practised in specific algorithmic contexts using concepts drawn from flowcharts, pseudocode and programming. The teacher support material illustrates examples such as the home/locker/knapsack for thinking ahead.

| | **Assessment statement** | **Explanations and Examples** |
|---|---|---|
| **Thinking procedurally** | | |
| 4.1.1 | Identify the procedure appropriate to solving a problem. | This means to write a set of steps for an ALGORITHM that solves a problem reliably. These could be written in **PSEUDOCODE** or in a FLOW-CHART.<br>For example: Add up all numbers from 1 to 99<br>SUM = 0<br>loop NUM from 1 to 99<br>SUM = SUM + NUM<br>end loop<br>output SUM |
| 4.1.2 | Evaluate whether the order in which activities are undertaken will result in the required outcome. | This means TESTING an algorithm using SAMPLE DATA.<br>For example, the following should add up all positive numbers in an array containing 5 numbers.<br>Test with NUMS = [ 5 , -5 , 0, -2 , 2 ]<br>loop C from 0 to 4<br>SUM = 0<br>SUM = SUM + NUMS[C] |

| | | |
|---|---|---|
| | | end loop<br>---> answer : result is SUM = 2<br>but it should produce SUM = 7<br>The SUM = 0 command should be before the loop command |
| 4.1.3 | Explain the role of sub-procedures in solving a problem. | If the programming language provides pre-written procedures,<br>then the algorithm can execute a pre-written command instead of the programmer writing lots of code.s<br>For example, if the language contains a SUM command for<br>adding up an array, then it's not necessary to write a loop.<br>Instead, just write : ANSWER = SUM(NUMBERS) |
| **Thinking logically** | | |
| 4.1.4 | Identify when decision-making is required in a specified situation. | In PSEUDOCODE, decisions are written as IF...THEN…<br>In FLOW-CHARTS, use a Diamond to represent a decision. |
| 4.1.5 | Identify the decisions required for the solution to a specified problem. | Common decisions are:<br>Strings : if ANSWER = "YES" then…<br>Numbers : if AGE > 21 then …<br>Boolean : if FOUND = TRUE then ... |
| 4.1.6 | Identify the condition associated with a given decision in a specified problem. | Compound BOOLEAN conditions us AND, OR, NOT<br>For example :<br>if AGE >= 13 AND AGE <=18 then<br>SCHOOL = "High School"<br>end if<br><br>if TODAY = "SAT" OR TODAY = "SUN" then<br>output "Weekend"<br>end if<br><br>if NOT( PASSWORD = "magic" ) then<br>output "Wrong Password"<br>end if |
| 4.1.7 | Explain the relationship between the decisions and conditions of a system. | For example:<br>if PASSWORD = "magic" then<br>CORRECT = TRUE<br>else if PASSWORD = "MAGIC" then<br>CORRECT = TRUE<br>else |

| | | |
|---|---|---|
| | | CORRECT = FALSE<br>end if |
| 4.1.8 | Deduce logical rules for real-world situations. | For example, in a blackjack game:<br><br>if MYTOTAL = 21 AND MYCARDCOUNT = 2 then<br>output "I win"<br>else if MYTOTAL > 21 then<br>output "I Lose"<br>else if DEALERTOTAL > 21 then<br>output "I Win"<br>else if MYTOTAL > DEALERTOTAL then<br>output "I Win"<br>else<br>output "I Lose"<br>end if |
| **Thinking ahead** | | |
| 4.1.9 | Identify the inputs and outputs required in a solution. | For example:<br>Algorithm should ADD UP all positive numbers in an array<br>Test Case A<br>Input = [ 5 , -2 , 0 , -5 , 3 ]<br>Output = SUM = 8<br>Test Case B<br>Input = [ -1 , -2 , -3 ]<br>Output = 0 |
| 4.1.10 | Identify pre-planning in a suggested problem and solution. | Gantt Chart - for project planning |
| 4.1.11 | Explain the need for pre-conditions when executing an algorithm. | Examples :<br>(a) Before comparing two Strings containing dates, the dates probably need to be in a specific format, like "dd-mm-yyyy"<br><br>(b) BINARY SEARCH can only be performed in an array if the array is already SORTED - so a precondition of is having a SORTED array<br><br>(c) Before converting a String containing a Binary number to decimal, the String must contain at least one character,<br>and all characters must be 0 or 1. |

| | | |
|---|---|---|
| 4.1.1 2 | Outline the pre- and post-conditions to a specified problem. | For example, if you want to BINARY SEARCH for and item in an array:<br>pre-condition : Array must be sorted before searching<br>post-condition : BINARY_SEARCH method must return the position of the item, if found, or -1 if it is not found |
| 4.1.1 3 | Identify exceptions that need to be considered in a specified problem solution. | For example:<br><br>(a) Solving Ax^2 + Bx + C = 0, there is NO SOLUTION if B^2-4AC < 0 - that's an exception<br><br>(b) When searching in a Linked-List, it could happen that the HEAD pointer is NULL. Then the search must end immediately, returning NOT FOUND |
| **Thinking concurrently** | | |
| 4.1.1 4 | Identify the parts of a solution that could be implemented concurrently. | For example :<br>We cannot SEARCH an array at the same time that a different algorithm is SORTING the array<br>But:<br>We can ADD UP all the numbers in an array at the same time that a different algorithm is searching for the largest and smallest values, and a third algorithm is copying all the values from the array into a TextArea on the computer screen. |
| 4.1.1 5 | Describe how concurrent processing can be used to solve a problem. | Algorithms are not required, but a Gantt chart would be appropriate. Concurrent processes are are the ones with bars that overlap. |
| 4.1.1 6 | Evaluate the decision to use concurrent processing in solving a problem. | Concurrent processing makes sense when it is possible to use multiple devices simultaneously, hence reducing the amount of time required.<br><br>Usually, WRITE commands cannot be done concurrently with anything else, whereas READ commands can be done concurrently. |
| **Thinking abstractly** | | |
| 4.1.1 7 | Identify examples of abstraction. | Abstraction means simplifying reality so that it can be represented (stored) inside a computer. For example,<br>a road map can be represented with codes for the starting |

| | | |
|---|---|---|
| | | and ending points of the road, plus a number that tells the length of the road. It is not necessary to actually have<br>a picture of the turns in the road. A famous problem is the Bridges of Koenigsberg, where only the starting and ending points are saved. |
| 4.1.1 8 | Explain why abstraction is required in the derivation of computational solutions for a specified situation. | In addition to encoding data items, for example as numbers,<br>DATA-STRUCTURES provide a chance to represent RELATIONSHIPS between data items. For example, an ARRAY or a LINKED-LIST can record data items in a specific order. A TREE can save a HIERARCHICAL STRUCTURE - for example representing a mathematical formula or a directory tree. But neither a LIST nor a TREE<br>can correctly represent the WEB, which has arbitrary numbers of links from any node, as well as links going in both directions between nodes. |
| 4.1.1 9 | Construct an abstraction from a specified situation. | This might be done using a DIAGRAM,<br>like a tree with lines linking nodes together.<br>Or a 2-dimensional array (table). |
| 4.1.2 0 | Distinguish between a real-world entity and its abstraction. | For example, an ID number for a student and the student's NAME are not the same as the actual student. Usually, abstractions only represent part of the important data. A student also has a birthdate, a phone number, height, weight, nationality, etc. Even with all that data, we still would not have a PICTURE of the students face, and we would not know<br>what they ate during the past week.<br><br>The ingredients shown on a cereal box might represent all the significant nutritional information, but they do not tell us<br>how the cereal will taste, or whether it is crunchy. |

**4.2 Connecting computational thinking and program design (22 hours)**

The focus of this topic is how an understanding of programming languages enhances the students' understanding of computational thinking and provides them with opportunities for practical, hands-on experience of applying computational thinking to practical outcomes.

In externally assessed components questions will be presented using flow charts and/or pseudocode as outlined in the approved notation sheet. Answers will only be required in pseudocode.

| | **Assessment statement** | **Ob** | **Teacher's notes** | **Explanations and Examples** |
|---|---|---|---|---|
| 4.2.1 | Describe the characteristics of standard algorithms on linear arrays. | 2 | These are: sequential search, binary search, bubble sort, selection sort. | Sequential Search<br>Start at the beginning<br>Check each position in the array<br>If target item is found then<br>output POSITION where item was found<br>else<br>move to next item<br><br>Binary Search<br>Start in the middle of the array<br>if target item is found then<br>output POSITION where item was found<br>else if target < current item then<br>start again in the middle of the first half of list<br>else<br>start again in the middle of the 2nd half of list<br><br>Bubble Sort<br>N = number of items in the array<br>Search through entire list N times<br>At each position, compare DATA[X] to DATA[X+1]<br>swap items X and X+1 if they are not in order<br>continue to end of list<br>Next pass<br><br>Selection Sort<br>find BEST item in list<br>swap it to beginning of list<br>find BEST item starting in second position<br>swap into 2nd position<br>repeat at 3rd position, then 4th position, etc<br>Requires N passes, but each pass is 1 item shorter |
| 4.2.2 | Outline the standard | 2 | These are: addition and retrieval of data. | A collection is an UNORDERED list of data - a "SET" |

| | | | | |
|---|---|---|---|---|
| | operations of collections. | | | The collection is a data-structure that has these operations:<br>.addItem( data )<br>.resetNext() = start at the beginning<br>.hasNext() → tells whether there is another item in the list<br>.getNext() → retrieves a data item from the collection<br>.isEmpty() → check whether collection is empty |
| 4.2.3 | Discuss an algorithm to solve a specific problem. | 3 | Students should be expected to discuss the differences between algorithms, including both standard and novel algorithms. For example, discussing the advantages and disadvantages of using a binary search as opposed to a sequential search. | A binary search is faster - O( log N ),<br>but can only be performed in a SORTED list<br>A sequential search is slower - O(N)<br>but can be performed whether the list is sorted or not<br><br>A bubble sort can "quit early" if no swaps are made<br>in a pass. But it makes lots of swaps.<br>A selection sort must always perform N passes -<br>it cannot "quit early". But it makes fewer swaps -<br>maximum of N swaps<br>Both Bubble and Selection sorts are O(n^2) |
| 4.2.4 | Analyse an algorithm presented as a flow chart. | 3 | Examination questions may involve variables, calculations, simple and nested loops, simple conditionals and multiple or nested conditionals. This would include tracing an algorithm as well as assessing its correctness.<br>Students will not be expected to construct a flow chart to represent an algorithm in an externally assessed component.<br>**MYP** Mathematics: using flow charts to solve problems in real-life | Be sure to learn the correct shapes of boxes:<br><br>Input/Output = parallelogram<br><br>Process(calculation) = rectangle<br><br>Decision (if..then) = diamond<br><br>Start/end/connect = circle<br><br>This is equivalent to the pseudocode:<br><br>ANSWER = input("Is it raining?")<br>if ANSWER = "yes" then<br>RESULT = "Take an umbrella"<br>else<br>RESULT = "Leave umbrella at home" |

| | | | | |
|---|---|---|---|---|
| | | | contexts, patterns and sequences, logic, algorithms.<br>**MYP** Technology: design cycle (inputs, processes, outputs, feedback, iteration). | end if<br>output RESULT |
| 4.2.5 | Analyse an algorithm presented as pseudocode. | 3 | Examination questions may involve variables, calculations, simple and nested loops, simple conditionals and multiple or nested conditionals. This would include tracing an algorithm as well as assessing its correctness.<br>**MYP** Mathematics: using flow charts to solve problems in real-life contexts, patterns and sequences, logic, algorithms.<br>**MYP** Technology: design cycle (inputs, processes, outputs, feedback, iteration). | Practice here: **Pseudocode Practice Tool**<br><br>Here are some old IB questions presented with Pseudocode.<br>Old Exam Questions<br>These questions are REALLY OLD and require some updating still. |
| 4.2.6 | Construct pseudocode to represent an algorithm. | 3 | **MYP** Mathematics: using flow charts to solve problems in real-life contexts, patterns and sequences, logic, algorithms.<br>**MYP** Technology: design cycle (inputs, processes, outputs, feedback, iteration).<br>**AIM 4** Demonstrate thinking skills to represent a possible solution to a specified complex problem. | Practice here: **Pseudocode Practice Tool** |
| 4.2 | Suggest suitable | 3 | Suitable algorithms may | For example: |

| | | | | |
|---|---|---|---|---|
| .7 | algorithms to solve a specific problem. | | include both standard algorithms and novel algorithms. Suitable may include considerations of efficiency, correctness, reliability and flexibility. Students are expected to suggest algorithms that will actually solve the problem successfully. **LINK** General principles of computational thinking, introduction to programming. | If PRICES and NAMES and INVENTORY are parallel arrays,<br>write an algorithm that finds all the items where<br>INVENTORY is below 10 items, and adds 20% to the<br>PRICES of those items. |
| 4.2.8 | Deduce the efficiency of an algorithm in the context of its use. | 3 | Students should understand and explain the difference in efficiency between a single loop, nested loops, a loop that ends when a condition is met or questions of similar complexity.<br>Students should also be able to suggest changes in an algorithm that would improve efficiency, for example, using a flag to stop a search immediately when an item is found, rather than continuing the search through the entire list. | Big O notation is not required, but makes things simpler:<br><br>Sequential Search → O(n) (n is the length of the list)<br><br>Binary Search → O( log n ) (that's log base 2)<br><br>Bubble Sort → O( n^2 )<br><br>[HL] Travelling Salesman Problem → O( n ! )<br>(n=number of cities) |
| 4.2.9 | Determine the number of times a step in an algorithm will be performed for given input data. | 3 | Examination questions will involve specific algorithms (in pseudocode/flowcharts), and students may be expected to give an actual number (or range of numbers) of iterations that a step will execute. | "number of steps" is called ITERATIONS<br><br>In nested loops, multiply the lengths of each loop<br>to determine iterations of inner-most command |

**4.3 introduction to programming (13 hours)**

| | Assessment statement | Ob | Teacher's notes | Explanations and Examples |
|---|---|---|---|---|
| **Nature of programming languages** | | | | Here are some good basic notes for this section: https://www.bbc.co.uk/education/guides/zgmpr82/revision/1<br><br>Here is a brief video about various levels of programming languages: https://www.youtube.com/watch?v=QdVFvsCWXrA |
| 4.3.1 | State the fundamental operations of a computer. | 1 | These include: add, compare, retrieve and store data.<br>Complex capabilities are composed of very large numbers of very simple operations. | ADD means numerical addition<br><br>This is referring to Machine Code - the language that<br>a CPU actually understands = NATIVE code |
| 4.3.2 | Distinguish between fundamental and compound operations of a computer. | 2 | For example, "find the largest" is a compound operation. | Add accumulator plus 1 → fundamental<br>Store accumulator into memory → fundamental<br>Storing 1,3,5,7,...,99 → compound (uses a loop)<br>Comparing two Strings → compound, as it must loop through<br>the Strings and compare many individual characters |
| 4.3.3 | Explain the essential features of a computer language. | 3 | For example, fixed vocabulary, unambiguous meaning, consistent grammar and syntax.<br>**TOK** Language and meaning. | English is not a computer language because meaning is<br>NOT unambiguous - for example "store" has several<br>different meanings<br><br>Example computer languages:<br>Java, Javascript, Python, Fortran, Basic, C++ |
| 4.3.4 | Explain the need for higher level languages. | 3 | For example, as the human needs for computer systems have expanded it is necessary | High Level code is easier to write, as one single command<br>like **output ARRAY** might print out 100 numbers, |

| | | | | |
|---|---|---|---|---|
| | | | to abstract from the basic operations of the computer. It would take far too long to write the type of systems needed today in machine code. | but low level languages require a loop for this task.<br><br>High Level code may provide more sophisticated concepts,<br>like Object Oriented Programming, that make programming<br>easier and quicker and more reliable<br><br>High Level language tools often provide sophisticated and<br>automated error-checking, -prevention, and -handling<br><br>High Level code also supplies LIBRARIES, such as<br>an Email CLASS that can send email messages |
| 4.3.5 | Outline the need for a translation process from a higher level language to machine executable code. | 2 | For example, compiler, interpreter, virtual machine. | CPUs and Compilers and Virtual Machines<br><br>Compiler - reads the entire program (source code) searching<br>for syntax errors. If no errors are found, it translates<br>the High Level source code into low level machine code,<br>that can run directly on the CPU.<br><br>Interpreter - reads one command at a time and then executes<br>it immediately. So if there is an error later in the program,<br>the program will start but fail in the middle. Interpreters<br>generally provide less debugging help than compilers.<br><br>A Virtual Machine is a software level that allows the compiler<br>or interpreter to produce output code that does not need<br>to run directly on the CPU, but rather in the Virtual Machine.<br>Java provides a virtual machine that is an |

| | | | | |
|---|---|---|---|---|
| | | | | identical<br>run-time environment on all platforms, so Java does not<br>need to be converted to native code. By creating a virtual<br>machine for each platform, Java is made "cross-platform"<br>by design. Unfortunately, this does not work as perfectly<br>as we might expect, due to minor hardware and OS<br>differences. |
| **Use of programming languages**<br>Sub-programmes and objects support abstraction, which facilitates: ease of debugging and maintenance, reuse of code, modularity.<br>There is no programming language specified in the SL/HL core. However, students must use a language that supports the basic constructs on the approved notation sheet<br>. | | | | |
| 4.3.6 | Define the terms: variable, constant, operator, object. | 1 | | variable = a name that represents a value<br>constant = a value that cannot change during run-time<br>operator = numerical operations, String operations,<br>logical (boolean) operations<br>e.g. operations on primitive data types<br>object = a collection of data and methods, created from<br>a design (class), allowing multiple INSTANCES<br>An object has a REFERENCE VARIABLE<br>that "points to" the contents of the object |
| 4.3.7 | Define the operators =, ≠, <, <=, >, >=, mod, div. | 1 | **LINK** Approved notation sheet. | div = integer division, no fractional part in the result<br>mod = the remainder from an integer division<br>The others are COMPARISON operators - |

| | | | | |
|---|---|---|---|---|
| | | | | they are also BOOLEAN operators as they produce TRUE or FALSE results. |
| 4.3.8 | Analyse the use of variables, constants and operators in algorithms. | 3 | For example, identify and justify the use of a constant as opposed to a variable in a given situation.<br>**MYP** Mathematics: forms of numbers, algebra—patterns and sequences, logic, algorithms. | Local variable = created inside a method, with SCOPE<br>that extends only to that method - it is not available<br>outside that method<br><br>Global variable = created outside all methods, and hence<br>usable (meaningful) in all the methods in the program |
| 4.3.9 | Construct algorithms using loops, branching. | 3 | Teachers must ensure algorithms use the symbols from the approved notation sheet.<br>**LINK** Approved notation sheet.<br>**MYP** Mathematics: using flow charts to solve problems in real-life contexts, logic, algorithms<br>**MYP** Technology: design cycle (inputs, processes, outputs, feedback, iteration).<br>**LINK** Connecting computational thinking and program design. | Loops - in Pseudocode : loop C from 1 to 10<br>...<br>end loop<br><br>loop while not FOUND do<br>…<br>end loop<br><br>Branching - in Pseudocode : if … then<br>…<br>else if … then<br>…<br>else<br>... |
| 4.3.10 | Describe the characteristics and applications of a collection. | 2 | Characteristics:<br>• Contains similar elements.<br>**LINK** HL extension, recursive thinking.<br>**LINK** General principles of computational thinking, connecting computational thinking and program design. | A COLLECTION is like a linked-list,<br>but the order of elements is not guaranteed.<br>Collection methods in Pseudocode are:<br>.addItem( new data item )<br>.resetNext( ) start at beginning of list<br>.hasNext( ) checks whether there are still more items in the list<br>.getNext( ) retrieve the next item in the list<br>.isEmpty( ) check whether the list is empty<br><br>Example: |

| | | | | |
|---|---|---|---|---|
| | | | | <pre>NAMES = new Collection()<br><br>NAMES.addItem("Bob")<br>NAMES.addItem("Dave")<br>NAMES.addItem("Betty")<br>NAMES.addItem("Kim")<br>NAMES.addItem("Debbie")<br>NAMES.addItem("Lucy")<br><br>NAMES.resetNext()<br><br>output "These names start with D"<br><br>loop while NAMES.hasNext()<br>  NAME = NAMES.getNext()<br>  if firstLetter(NAME) = "D" then<br>  output NAME<br>  end if<br>end loop<br><br>method firstLetter(s)<br>  return s.substring(0,1)<br>end method</pre><br>Output:<br>Dave<br>Debbie |
| 4.3.11 | Construct algorithms using the access methods of a collection. | 3 | **LINK** Connecting computational thinking and program design. | See the examples in **Pseudocode Practice Tool** |
| 4.3.12 | Discuss the need for sub-programmes and collections within programmed solutions. | 3 | Show an understanding of the usefulness of reusable code and program organization for the individual programmer, team members and future maintenance. | Too bad that IB Pseudocode contains no facility for writing methods or sub-programs.<br><br>In any case, we use subprograms (methods, functions)<br>for a several reasons:<br>(1) it permits RE-USING code, rather than copying and pasting |

| | | | | |
|---|---|---|---|---|
| | | | **LINK** General principles of computational thinking, connecting computational thinking and program design.<br>**MYP** Technology: use of software such as Alice. | (2) calling with PARAMETERS allows the sub-program to<br>perform a variety of different tasks<br>(3) once the sub-program has been DEBUGGED (tested and fixed), it can be re-used by multiple programmers and hence improve productivity<br>(4) encourage multiple programmers in a team to reuse<br>the same sub-programs improves reliability<br>(5) when changes are needed, changing one sub-program<br>is much easier, faster and more reliable than making changes<br>in multiple copies of similar code |
| 4.3.13 | Construct algorithms using predefined sub-programmes, one-dimensional arrays and/or collections. | 3 | **MYP** Mathematics: using flow charts to solve problems in real-life contexts, logic, algorithms.<br>**MYP** Technology: design cycle (inputs, processes, outputs, feedback, iteration); use of software such as Alice. Students will only be required to analyse flow charts in the externally assessed components. Students will be expected to write and analyse pseudocode in the externally assessed components.<br>**S/E**, **AIM 8** Appreciate the implications of using available code from sources such as online forums.<br>**LINK** Connecting computational thinking and program design. | See the examples in<br>**Pseudocode Practice Tool**<br><br>Although Alice is mentioned here, it is not necessary to<br>use or master Alice. Java is a better example language.<br>Exam questions will always be presented and answered<br>in the IBO Pseudocode system. |

# Topic 5: Abstract Data Structures

### 5.1.1 Identify a situation that requires the use of recursive thinking

**Recursion:** a method where the solution to a problem depends on solutions to smaller instances of the same problem. OR a method that calls itself

**Towers of Hanoi**

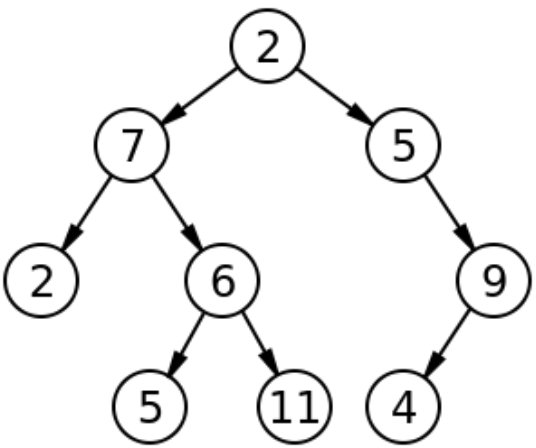
- The objective of the puzzle is to move the entire stack to another rod, obeying the following rules:

- Only one disk may be moved at a time.
- Each move consists of taking the upper disk from one of the rods and sliding it onto another rod, on top of the other disks that may already be present on that rod.
- No disk may be placed on top of a smaller disk.

**Step 1 - Identify the recursive call**
**1.** To move all but one disk to the spare peg. This can be done recursively.
**2.** Then move the bottom disk to the destination peg.
**3.** Move all the other disks onto the destination peg. (This can be done recursively)
**Step 2 - Identify the base case**
The base case is when the base disk has nothing.
So in pseudocode this would look like this:

```
HANOI ( N , BEGINNINGPEG, SPAREPEG, ENDPEG) {
  if N=1 then
    move BEGINNINGPEG to ENDPEG
  end if
  else
    HANOI (N-1, BEGINNINGPEG, ENDPEG, SPAREPEG)
    HANOI (1, BEGINNINGPEG, SPAREPEG, ENDPEG)
    HANOI (N-1, SPAREPEG, BEGINNINGPEG, ENDPEG)
  end else
```

**Parse binary trees**




In computer science, a binary tree is a tree data structure in which each node has at most two

child nodes, usually distinguished as "left" and "right".

Queues: The primary use of a queue is to ensure the service that is being queued for is given first to those that joined the queue first. This is known as a First In First Out structure (FIFO). Some examples of how queues are used in computer science are as follows:

1. ***Keyboard queue****:* The keyboard buffer into which characters are stored as they are pressed operates as a queue. The first letter typed is the first letter sent. Subsequent letters are added at the tail of the buffer.
2. ***Printer queues****:* Requests for printing are sent and the fist sent is the first printed. A printer queue stores the requests as they arrive.

To get the array to act as a stack we need to have some way of keeping track of which element contains the topmost item. This can be easily achieved using an int variable (“top” in the diagrams). An empty stack could be represented by -1 (zero would be a stack with one item). Attempting to pop an item off an empty stack would result in a **stack underflow**. Similarly, attempting to push an item onto a full stack results in **stack overflow**.

There are two ways of implementing a queue using an array – a linear queue and a circular queue.

**Linear queues:** easier to visualize and code
Circular queues: more efficient and avoid need to move data along the array as items are dequeued from the head.

Many strategies to implement array as linear queue. One:

1. The head of the queue is always element zero
2. The tail of the queue grows from zero as the queue is filled
3. full queue is signaled by tail being the last element in the queue.
4. Items are always dequeued from array element zero (head). Once an item has been dequeued, two things must happen:
    a. All other items must be shuffled forward by one element
    b. The variable that stores tailMarker must be decremented to reflect the new position of the tail of the queue.

**Dynamic and Static Data Structures**

| Dynamic data structure | Static data structure |
|---|---|
| Memory is allocated to the data structure dynamically as the program runs. | Memory size is fixed and set at the time of compiling. |
| Disadvantage. There may be overflow or underflow during runtime if allocations are exceeded. | Advantage.Memory size is fixed and so there will be no problems during run time with memory allocations |
| Advantage. Makes the most efficient use of memory as it is using only as much as it needs | Disadvantage. As memory size is fixed there may be a lot of wasted memory space during the running of the program. |
| Disadvantage. Harder to program as the program needs to keep track of data size and locations during the running of the program | Advantage. Easier to program as there is no need to check on memory size or make use of algorithms that require links to new data. |

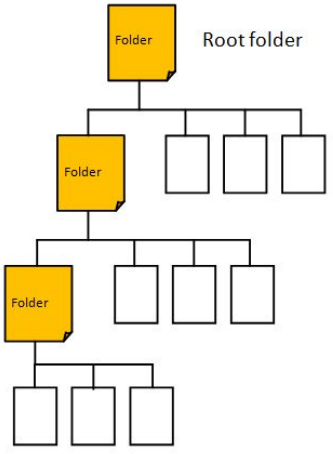
**Pointers required for Linked Lists:**

1. The start or head pointer points to the first element in the list.
2. The last node has a null pointer which is an empty pointer.
3. Each other node has a pointer pointing to the next node in the list.
4. In order to access the list we need to always start at the head.
5. To move through the list we follow the pointers to the end until we encounter the null pointer.
6. In a singly linked list we can only move one way through the list.

**Binary Trees**

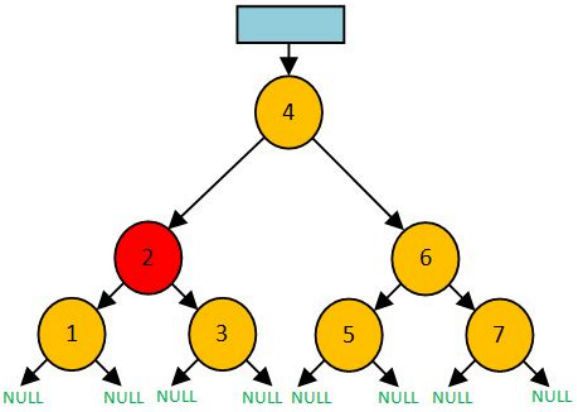
- Tree structures are very common in computing, the most familiar perhaps being the organisation of files and folders (or directories if you prefer) on a storage device.

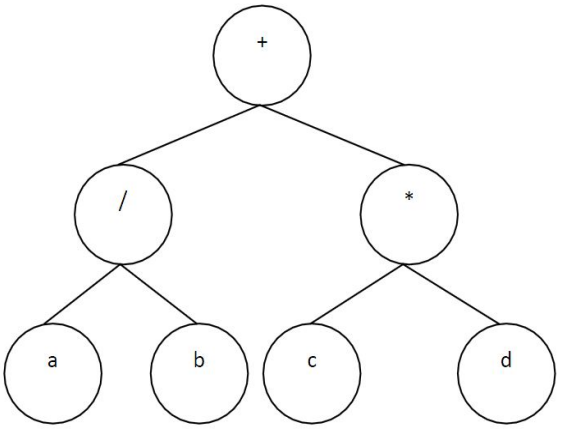
1. each folder contains one sub-folder and three files, except the lowest level folder, which only contains files.
2. Most efficient way to conduct operation in all folders: recursion.




```
public void setAttrib (Folder thisFolder)
{
        if(hasSubFolder(thisFolder))
        {
                setAttrib(thisFolder.subFolder);
        }
        setFiles(thisFolder,ro);
}
```

- method would initially be called with the root (top level) folder passed as a parameter. Since hasSubFolder would evaluate to true, the recursive call on line 5 to setAttrib will execute, passing the subfolder as a parameter. The first call will be pushed onto a stack.
- Line 5 will execute again and make another call to setAttrib, passing the third level folder as a parameter. The second call will be pushed onto a stack.
- The third level folder has no subfolder, so this call will jump to line 7, set the attributes of all files in the third level folder to read-only and then complete.
- The second call will be popped of the stack and the algorithm continued from where it was interrupted by the recursive call (at the end of line 5). Line 7 will execute and this call will complete.
- Finally, the original call will be popped off the stack and resumed, executing line 7 and completing.
- So we now have all the files in all the folders processed, all calls completed and an empty stack.




- Node 2 is the *child* node of 4, but the *parent* node of 1 and 3.
- 2 is in 4's left *subtree*.
- It is the *root node* of two subtrees, a *left subtree* containing 1 and a *right subtree* containing 3.
- It is a *sibling* node to 6 (sibling nodes are at the same level in the tree).
- 2's two child nodes (1 and 3) are both *leaf nodes* as they have no subtrees.

**In order traversal** would result in infix. Rule: A node is visited in-between its left subtree and right subtree (Left visited first).
a/b+c*d
**Preorder traversal would result in prefix**. Rule: A node is visited before its descendants.
+/ab*cd
**Postorder traversal would result in postfix**. Rule: A node is visited after its descendants.
ab/cd*+

**Dynamic Data:** Dynamic data is data that changes when further updates to the data become available.

**5.1.20 Suggest a suitable structure for a given situation.**

There are many situations where the size of the structure is not known beforehand. These cases lend themselves to dynamic data structures. An example of this is the programming of a printer spooler where the number of print jobs will vary and is not known beforehand. Other situations may occur where the size of the data structure can be determined beforehand in this case a static data structure would be used.

**Vocabulary**

- **Root** - Base Node
- **Child** - Node with directed edge towards it
- **Parent** - Node with directed edge out of it
- **Sibling** - Node that shares a parent with another
- **Descendant** - Child's Child any number of non-zero times
- **Ancestor** - Parent's Parent any number of non-zero times
- **Leaf** - Node with no children
- **Internal Node** - Node with children
- **External Node** - Leaf
- **Degree** - Number of subtrees of a node
- **Edge** - Connection between nodes
- **Path** - List of adjacent nodes
- **Level** - Distance from root + 1
- **Height of Node** - Distance from root
- **Height of Tree** - Height of furthest leaf
- **Depth** - Distance from root
- **Forest** - Group of disconnected trees

# Topic 7: Control

1. Washing Machine:
    a. The cycle switch has the job of determining how long the different parts of the cycle last.
    b. Inside the switch: motor equipped with a gear reduction that makes the control dial turn very slowly.
        i. Set of six contacts, actuated by the small pieces of metal in the plastic arm on the dial.
        ii. As the dial spins, bumps on the dial raise and lower the six metal pieces, which close and open the contacts in the top half of the switch.

   c. The length of the bumps determines how long each part of the cycle lasts, and the length of the space between bumps determines how long the machine pauses before moving on to its next task.
   d. These switches control the speed of the motor and determine which of the hot/cold water supply will open during the wash and the rinse cycles.
   e. The level sensor uses a pressure switch to detect the water level in the tub, which controls how high the tub fills with water.

2. Automatic Doors
   a. Automatic doors consist of a sensor, a processor and an automator.
   b. Sensor often infrared, sending out infrared signals which bounce off objects.
   c. The sensor then receives the signal and then sends a signal to the processing unit which then processes the information and sends it to the automator, in the form of two motors that cause the doors to open for a brief moment, and then close.

**Microprocessors**: A microprocessor (a.k.a. CPU) is a computing engine assembled in one chip. It performs all the calculations and data processing of the computer.

1. Advantages:
   a. Fast data processing speeds
   b. Quick reaction capability
   c. Control systems can run throughout the year 24/7
   d. They can work in places where it would be dangerous for a human to
   e. Outputs are consistent and error free
2. Disadvantages:
   a. Cost of specialized systems
   b. Lack of power - system off
   c. Computer malfunction - system off
   d. Programmability of limited scenarios
   e. Ethical concern - cannot risk total handover to one computer

**Sensors:** input devices that measure physical quantities of analogue data. Eg. temperature sensor, ambient light sensors, metal detectors

**Analogue to Digital Converter (ADC):** converts analogue data, from sensors, to readable data for a computer, which is digital data. A computer can't read analogue data because it is constantly changing, so and ADC will sample data, losing a lot of quality but making it possible for a computer to process it.

### 7.1.3 Evaluate different input devices for the collection of data in specified situations

**Sensors**

**Microsoft Kinect Sensor**

- Contains 3D depth sensors, colour camera and microphone array. Used in conjunction with XBox or Windows OS for motion sensing and voice control

**Air-Fuel Ratio Meter**

- Used in internal combustion engines (ICE) to determine how much oxygen is being outputted by the engine.
- Indicates whether all oxygen is being used by the engine and whether there is an engine malfunction
- Used to optimize fuel efficiency of an engine.

**UAV**

- Contain many kinds of sensors such as biological sensors which can detect the presence of various microorganisms in the air and other biological factors.
- Contain electromagnetic spectrum sensors which can detect visible light and infrared or near infrared as well as rad

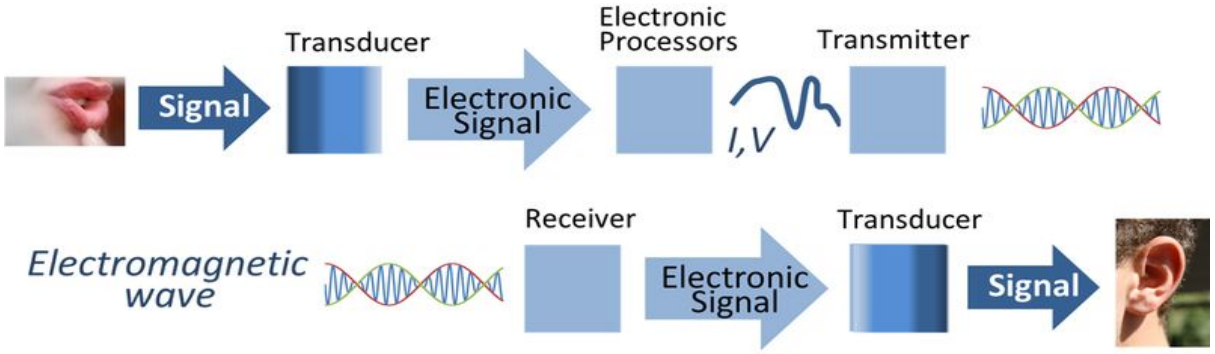
**7.1.4 Explain the relationship between a sensor, the processor and an output transducer.**

**Sensor Definition:** A device that detects or measures a physical property and records, indicates, or otherwise responds to it.

**Processor Definition:** A processor is the logic circuitry that responds to and processes the basic instructions that drive a computer.

**Transducer:** A substance or device, such as a piezoelectric crystal, microphone, or photoelectric cell, that converts input energy of one form into output energy of another.

**Relationship**




**7.1.5 Describe the role of feedback in a control system.**

**What is feedback?**

- "Information about reactions to a product, a person's performance of a task, etc., used as a basis for improvement."
- Feedback occurs when the outputs from a control system change the inputs, usually to keep the system in a stable state.

**How does feedback work?**

Consider a refrigeration system which is used to keep food frozen at below -20°C. A temperature sensor is used to measure the temperature inside the freezer. The system contains a refrigeration unit to cool the freezer. If the temperature is below -20°C then the refrigeration unit is turned off to save energy. If the temperature rises above -20°C then the refrigeration unit is turned on to cool the food down.

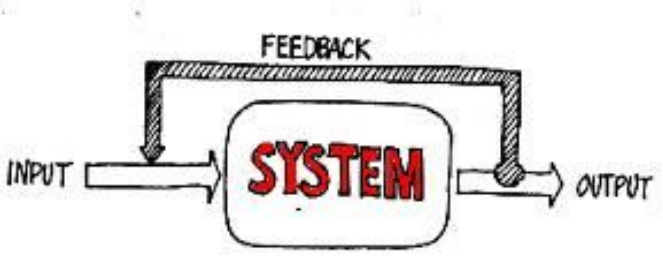
The cooling system inside the freezer will repeatedly follow this sequence of actions :

1. The input from the temperature sensor causes the refrigeration unit to turn on because the temperature is above -20°C.
2. The refrigeration unit is turned on so the temperature falls.
3. The temperature becomes so cold that the temperature sensor causes the refrigeration unit to be turned off.
4. The refrigeration unit is off so the temperature in the freezer rises slowly.
5. The system goes back to step 1.

In this system the inputs first affect the outputs (temperature sensor causes refrigeration unit to turn on). Then the outputs affect the inputs (refrigeration unit changes temperature reading). Because the outputs affect the inputs we can say that the system involves feedback. The feedback keeps the system in a stable (cold) state.

Feedback can occur in information-based systems as well. Often an output will have a result on further inputs. For example, the output of accepting an online booking for an air ticket will be to reduce the number of tickets available.

**Feedback loop**




**7.1.6 Discuss the social impacts and ethical considerations associated with the use of embedded systems**

One quite simple example is a car. Computer chips in cars improve fuel economy and control several parts of the car, including warning lights. The main issue related to this is that not all car engineers are then able to repair these cars once a problem is discovered. It basically requires a degree in electrical engineering to be able to repair a car that has a computer chip.

**Reliability and Integrity**

Some systems, like the cars mentioned above, may fail with the use of microchips. Because these chips are used, repairing the device or machine is much more difficult.

**Security**

There is a security issue regarding the use of embedded systems as these systems may have access to important information like credit card numbers etc. making it possible for potential hackers to access this information.

**People and Machines**

When people begin to rely on the use of IT in their everyday life, they sometimes become addicted to it and need it. This causes some people to perform less tasks than they could have. Certain tasks can be performed by computers, including tasks that could have been better by a human than a machine.

**7.1.7 Compare a centrally controlled system with a distributed system.**

SETI@Home

SETI@home uses distributed to perform data analysis on microwave signals received from space in the search for extraterrestrial intelligence. By installing a program (BOINC) on your computer you are able to become connected to a network of people's home computers which are in turn connected to a server where the data is collated. Currently there are over 3 million computers connected into this network.

**Climateprediction.net**

- Distributed computing project that works towards reducing the amount of uncertainties in climate models.
- Does this by running hundreds of thousands of different models, using the idle time of personal computers.
- Enables better understanding of how climate models are affected by changes in many parameters.
- Relies on volunteer computing, running client-side processes in people's computers.
- Outputs examined server-side; run mainly at Oxford, more data generated than any other model.

**Bitcoins**

- Digital currency described as a peer-to-peer, electronic cash system.
- Bitcoin creation and transfer is based on an open source cryptographic protocol and not managed by central authority.
- Bitcoin subdivided down to eight decimal places, forming 100 million smaller units called satoshis.
- Can be transferred through a computer or smartphone without an intermediate financial institution.
- Transaction processing is secured by servers called Bitcoin miners.
    - Communicate over an internet-based network and confirm transactions by adding them to a ledger, updated and archived periodically.
    - Each new ledger update creates some newly-mined bitcoins.
        - The number of new bitcoins created in each update is halved every 4 years until the year 2140 when this number will round down to zero.

**Rosetta@home**

- Distributed computing project for protein structure prediction on the Berkeley Open Infrastructure for Network Computing (BOINC) platform, run by the Baker laboratory at the University of Washington.
- Aims to predict protein–protein docking and design new proteins with the help of about sixty thousand active volunteered computers processing at 62 teraFLOPS on average as of October 18, 2011.
    - Also capable of researching Alzheimer's, Anthrax, HIV and malaria.
- Run through BOINC, available on Windows, MAC, Linux
    - Requirements: 500 mhz processing, 200MB free disk space, 512 MB physical memory, and Internet connectivity.

**Big and Ugly Rendering Project**

- 2D and 3D rendering for animation artists
- The idea of BURP is to use spare CPU cycles on participating computers around the world to render 3D images and animations submitted by the users of the BURP network.

**Open Rendering Environment (ORE)**

- Every user has access to the rendering content.
- The Berkeley Open Infrastructure for Network Computing (BOINC) is an open source middleware system for volunteer and grid computing.
- BOINC is software that can use the unused CPU and GPU cycles on a computer to do scientific computing.

**Pros and Cons of Distributed Computing**

**Pros:**

1. Cheaper than supercomputing
2. If sufficient computers are connected, performance exceeds supercomputing
3. Reliability: a few devices going offline does not impact on overall processing
4. Very easily scalable due to software nature

**Cons:**

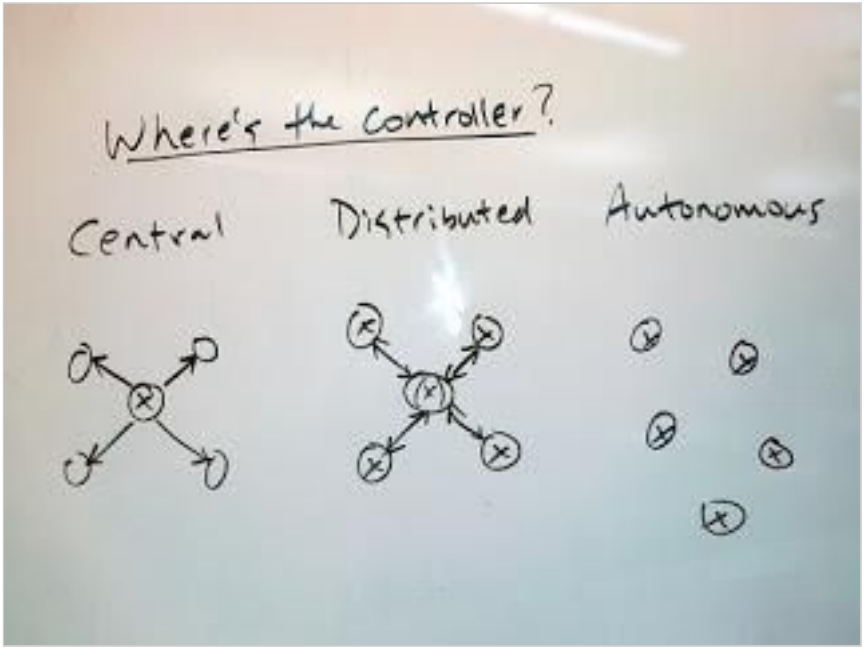
1. Bottlenecking can happen if a lot of data is being transferred at once
2. Security- potentially sensitive information can be leaked due to security breach
3. Software has to be written for the project, which can put people off of it if comes across as too complicated

### 7.1.8 Outline the Role of Autonomous Agents within a Larger System

| | **Advantages** | **Disadvantages** |
|---|---|---|
| **Centralized Control** | More control without need to search for problem or physically fix them | Expensive and complex. If central control fails, entire system is dead. |
| **Distributed Control** | If one part fails, the whole doesn't, exception: assembly lines | Not easy to isolate problems and then fix them as needed |
| **Autonomous Agents** | Possibly individually cheap, no complex control required | Agents need their own expensive controller. Since they're not connected, they are less efficient than when they are connected. |

**Examples**

1. **Centralized Control: Heating system at a school like ISP**
2. **Distributed Control: Traffic lights system with "smart" intersection traffic lights**
3. **Autonomous Agents: Swarming oil slick cleanup robots**



**http://ibcomp.fis.edu/2017/2017ibCompBlogyr2-2018.html**
**http://ibcomp.fis.edu/review/IBCS2018.html**

# Option A: Databases

**DEFINITIONS:**

1. **Dependency:** a dependent or subordinate thing, especially a country or province controlled by another.
2. **Transitive dependency:** transitive dependencies are when a nonkey attribute determines another attribute.
3. **Normalization:** Database normalization, or simply normalization, is the process of restructuring a relational database in accordance with a series of so-called normal forms in order to reduce data redundancy and improve data integrity.
4. **Transaction:** A sequence of database operations that access the database. A transaction is a logical unit of work that is it must be completed in its entirely or not at all.
5. **Schema:** representation of plan or idea in the form of model or outline.
6. **Referential Integrity Rule:** A condition by which a dependent table's foreign key must have either a null value or a matching entry in the relational table.

**A.1.1 Difference between Data and Information**

7. **Data:** facts and statistics collected together for reference or analysis.
8. **Information:** Information is stimuli that has meaning in some context for its receiver.

**A.1.2 Difference between Information System & Database**

9. **Database:** a structured set of data held in a computer, especially one that is accessible in various ways.
10. **Information System:** a computer system or set of components for collecting, creating, storing, processing, and distributing information, typically including hardware and software, system users, and the data itself: the use of information systems to solve business problems

**Data Warehouses**: repository of data collected from multiple data sources and intended to be used as a whole under the same unified schema

- option to analyze data from different sources under the same roof

- data warehouses are usually modeled by a multi-dimensional data structure

**Data mining**: (aka Knowledge discovery in databases): non trivial extraction of implicit, previously unknown and potentially useful information from data in databases (data mining is actually part of the KDD)

data mining algorithms can take advantage of the structure; benefits from sql for data selection, transformation, consolidation; but also goes beyond with predicting, comparing, detecting deviations, etc.

- descriptive data mining: describe the general properties of existing data
- predictive data mining: attempt to do predictions based on inference on available data
- **Characterization**: summarization of general features of objects in a target class; produce characteristic rules
  - relevant data are retrieved through a database query and run through summarization module to extract essence of data
- **Discrimination**
  - produces discriminant rules, comparison of general feature of objects between two classes (target and contrasting)
  - data discrimination uses comparative measures as well as stuff from characterization techniques
- **Association analysis**
  - discovery of association rules: frequency of items occurring together in transactional databases
  - threshold called support used to identify frequent item sets
  - additionally, confidence is the conditional probability that an item appears when another item appears, is used
- **Classification**
  - organization of data into given classes
  - aka supervised classification, uses given class labels to order objects in the data collection
  - normally use a training set where all objects are already associated
  - learning algorithm to figure stuff out and build a model
- **Prediction**
  - two types: prediction of unavailable data values or pending trends, or predicting a class label for data
  - prediction also refers to forecast of missing numerical values, or increase/decrease trends in time related data
- **Clustering**
  - organization of data into classes; however class labels are unknown and the algorithm must discover acceptable classes
  - aka unsupervised classification, based on maximizing similarity of objects in a class (intra-class similarity) and minimizing similarity in objects of different classes (inter-class similarity)
- **Outlier Analysis**

        - outliers are data elements that cannot be grouped in a class/cluster
        - while outliers may be considered noise, they can also reveal important information about other stuff
    - **Evolution and deviation analysis**
        - study of time related data that changes in time (lol)
        - models evolutionary trends in data
        - deviation analysis considers differences between measured values and expected values and attempts to find the cause in deviations
- usefulness of data
    - data mining produces a large number of patterns and rules; some consider meta-mining to mine from large data-mining results
    - we only want to consider what we deem interesting, so our measurement of interestingness is based off of validity when tested on new data, or other descriptions like understandability, novelty, or usefulness
    - also look at if they confirm or validate a hypothesis, or contradict a common belief
- categorization of data mining systems
    - according to type of data mined: categorizes systems according to the type of data handled
    - data model drawn on: based on the data model involved
    - kind of knowledge discovered: based on knowledge discovered or data mining functionalities
    - mining techniques used: based on data analysis approach used
- ISSUES IN DATA MINING
    - **Security/social**
        - large amounts of sensitive and private information about individuals or companies
        - could disclose new implicit knowledge about individuals or groups that could violate privacy policies
        - databases are often sold, and some information may be withheld while others may be widely distributed
    - **User Interface**
        - knowledge discovered must be understandable by the user
        - good data visualization eases interpretation and helps users understand their needs
        - manage screen real estate, information rendering, interaction
    - **Mining methodology**
        - versatility of mining approaches, diversity of data available, dimensionality of domain, broad analysis needs, assessment of knowledge discovered, exploitation of background knowledge and metadata, control and handling of noise in data, etc.
        - most algorithms consider data to be noise-free which often is not true; necessitates data preprocessing
        - size of search space must be handled also: curse of dimensionality (the more dimensions, the greater the space)
    - **Performance**

        - artificial intelligence and statistical methods were not designed for very large data sets
        - scalability and efficiency problems with large data
        - linear-time complexity algorithms are relied on, as well as sampling of a dataset
        - incremental updating and parallel programming
    - **Data source**
        - practical issues like diversity of data types, as well as philosophical like data glut (2 much data 4 me)
        - the more we use data mining, the more data we harvest, and it's way easier to collect data than it is to process it
        - heterogeneous databases and diverse complex data types: we cannot expect data mining systems to effectively and efficiently achieve good mining results on all kinds of data and sources

## LEVELS OF DATABASE ARCHITECTURE:

1. **Physical Data Level:** The **physical schema** describes details of how data is stored: files, indices, etc. on the random access disk system.
2. **Logical Data Level:** What data are stored in the database and What relationships exist among those data;  hides low level complexities of physical storage.
3. **External Data Level:** describes only part of the entire database that a end user concern; End users need to access only part of the database rather than entire database; highest level of data abstraction.

## PROPERTIES OF A TRANSACTION:

1. **Atomicity-** a transaction is treated as a single indivisible unit of work
2. **Consistency-** indicates the permanence of the database consistent state
3. **Isolation –** assures that the database used during the execution of a transaction cannot be used by a second transaction until the first one is completed
4. **Durability-** assures that once transaction changes are done , they cannot be undone or lost, even if the event in the event of a system failure
5. **Serializability-** ensures that the selected order of transaction operations creates a final database state that would have been produced if the transactions had been executed in a serial fashion.

## NORMALIZATION:

1. process of crystallizing entities and their relationships in the form of a relation: the theory of normalization allows us to recognize undesirable relations
2. Minimality - the minimum amount of items needed to make the key unique, candidate keys have this attribute
3. Composite keys: composed of multiple columns
4. organization of a database into columns (attributes) and tables (relations) to eliminate data redundancy and improve data integrity

5. normalization theory is built around normal-forms: a relation is of a particular form if it satisfies a set of constraints
6. an unnormalized relation is one in which an element in a row column position is not atomic, i.e. there is more than one value in that position
7. **Types of anomalies:**
    a. **Insert**: unable to record certain facts until the entire row is known, part of the key is missing
    b. **Update**: the same information is recorded in multiple rows, so updating one row does not update info in all rows
    c. **Deletion**: deletion of a row causes more than one set of facts to be removed
8. **First Normal form - Every element can be accessed by a key, only one value for a row/column position**
    a. transform data items to a two dimensional table; remove repeating groups
    b. a relation is in first normal form if and only if all underlying simple domains contain atomic values only
        i. The information is stored in a relational table and each column contains atomic values, and there are not repeating groups of columns.
        ii. **atomic**: columns cannot contain sub-columns (values in columns cannot be further subdivided, e.g. "Chicago", not "Chicago; Los Angeles")
    c. for a given value of the particular key, the values of the non-key attribute cannot be determined uniquely
    d. problems:
        i. insertion: cannot insert a new row if missing part of the key
        ii. deletion: deleting part of the tuple of a patient can cause problem
        iii. update: redundancy issues
        iv. some attributes depend on only part of the primary key, but the entire primary key is necessary to identify a tuple
9. **Second Normal Form - All of the attributes describe the key**
    a. a relation is in 2nd normal form iff it is in 1st normal form and every nonkey attribute is fully dependent on the primary key
        i. The table is in first normal form and all the columns depend on the table's primary key.
        ii. for each column, "Does this column serve to describe what the primary key identifies?" e.g. key "Employee", all columns should describe an "Employee"
    b. still problems:
        i. insertion: "cannot enter the fact that a particular drug has a particular side effect unless the drug has been given to the patient"; "cannot enter a tuple in the relation unless the patient has been operated upon and given a drug"

   ii. deletion: non-key attribute is partially dependent on another non-key attribute
   iii. update: need to search for occurrences in relation in order to update
      1. inconsistencies in the relation compromise the integrity of the database
   iv. dependencies of non-key attributes, aka transitive dependencies

**10. Third Normal Form - None of the attributes describe each other**

   a. a relation is in 3rd normal form iff it is in 2nd normal and every nonkey attribute is non-transitively dependent on the primary key
      i. the table is in second normal form and all of its columns are not transitively dependent on the primary key
      **ii.** **transitive dependence**: a column's value relies upon another column through a second intermediate column
         1. e.g., field "AuthorNationality" is transitively dependent on "Author", which are both dependent on the key "Book"
   b. non-key attributes are mutually independent, and fully dependent on the primary key
   c. a non-key attribute is any attribute that does not participate in the primary key of the relation concerned
   d. two or more attributes are mutually independent if none of them are partially dependent on any combination of the other
   e. no data should be lost in the normalization process and it should be possible to move backwards
   f. not a function of data values that happen to be in the relation, but a function of the relationship between attributes

11. A relationship is a named association between entity types
   a. after completing normalization there are a set of entities and attributes
   b. relationships can be categorized as one-to-one, one-to-many, and many-to-many

12. Integrity: maintain accuracy and correctness of data in a database system
   a. integrity constraints are a set of rules or conditions to which all correct states of the database are required to satisfy
   b. integrity can be compromised by..
      i. poor validity checking - typing errors, logical errors, errors in software
      ii. data redundancy
   c. should monitor transactions to database tables and take action if integrity will be violated
   d. Domain Integrity rules
      i. domain integrity - type of domain, must be precise
      ii. maintain correctness of attribute values within relations, ensure values of a particular attribute are correct
   e. entity integrity rules

i. components of a primary key cannot have a null or missing value
    1. primary keys are used to uniquely identify a tuple in a relation
    2. entities that are meant to represent some real world feature must be distinguishable

ii. The primary key for a row is unique; it does not match the primary key of any other row in the table.

f. Intra Relational integrity
    i. maintain correctness of relationships among attributes of a relation, esp. key uniqueness
    ii. no two tuples in a relation have the same values for those attributes composing the primary key

g. referential integrity rules
    i. correctness of relationships between relations
    ii. must not contain unmatched foreign key values
    iii. A condition by which the dependent table's foreign key must have either a null value or a matching entry in the relational table

h. Restricted effect: restrict delete operation to cases where there are no matching tuples with a foreign key reference
    i. restrict update to cases in which there are no matching tuples with a foreign key reference, otherwise don't update

i. cascade effect: delete tuple and all matching foreign key references
    i. update primary key and all matching foreign key references

j. nullify effect: set all foreign key references to null and delete the tuple ; set all matching foreign key references to null then update the tuple

13. data validation and verification
    a. when an attribute is added to the table it should be checked to ensure the data are members of the correct domain
    b. use range checks
14. disadvantages of relational databases
    a. recursive queries
    b. semantic knowledge - incorporates little semantic integrity, constraints must be explicit
    c. missing information
    d. no support for complex objects

**RELATIONSHIPS:**

- An **entity** is something about, which data are stored that is important to someone or something. An entity can be a person, place, concept, or event
- An **attribute** is a characteristic of an entity.
- A relationship describes an association among entities eg. an agent can serve many customers and each customer may be served by one agent.
- Three types of relationships are:
    - 1:1 - one-to-one: One bed, One person

- 1:M - one-to-many: One painter, Many paintings
- M:M - many-to-many: Many students, Many courses

**Data Modelling**

Data Modelling is the first step in the process of database design. This step is sometimes considered as a high-level and abstract design phase (conceptual design).

The aims of this phase is to:

- **Describe** what data is contained in the database (e.g. entities: students, lecturers, courses, subjects etc.)
- **Describe** the relationships between data items (e.g. Students are supervised by Lecturers; Lecturers teach Courses )
- **Describe** the constraints on data (e.g. Student Number has exactly 8 digits; a subject has 4 or 6 unit of credits only)

- The data items, the relationships and constraints are all expressed using the concepts provided by the high-level data model.

a. **One-to-one (1:1) relationship:** In a hospital environment health regulations may require that a bed be occupied by one patient. In turn a patient may occupy a single bed.
b. **One-to-one (1:M) relationship:** A painter paints many different paintings but each one is painted by the one painter.
c. **Many-to-many (M:M or M:N) relationship:** In a college, a student may take many courses and given any one course, many students are taking that course.

- A **constraint** is a restriction that is placed on the data. Constraints are important because they help to ensure data integrity. Constraints are expressed in the form of rules.
- A **business rule** is a brief, precise, and unambiguous description of a policy, procedure, or principle within a specific organization.

**TRANSACTION MANAGEMENT AND CONCURRENCY CONTROL:**

- **Transaction**
  - an action that reads from/writes to a database
  - insert (generates a list), update (change values), insert(add row), combination
  - represents a real world event (e.g. sale of a product)
  - must be a logical unit of work
    - takes a database from one consistent state to another - all parts of a transaction must be executed, otherwise abort
  - to avoid an inconsistent state, DBMS ensures database operations are completed before they are committed
- Transactions have four properties (**AIDS**):
  - **Atomicity** - all parts of a transaction must be executed, else it is aborted

  - **Isolation** - data used by one transaction cannot be used by another transaction until the first is completed (multi-user only)
  - **Durability** - once a transaction is committed, it cannot be rolled back
  - **Serializability** - result of concurrent execution of transactions is the same as if they were executed in serial (consecutive) order (multi-user only)
- **Concurrency Control**
  - act of coordinating simultaneous execution of transactions in a multiprocessing or multi-user database management system
  - ensures the serializability of transactions in a multi-user DBMS (DBMS scheduler is in charge of maintaining concurrency control)
  - helps guarantee data integrity and consistency in a database system

- **THREE PROBLEMS WITH CONCURRENT TRANSACTION EXECUTION**
  - **Lost updates:** successful and completed transactions can be overridden by another transaction
  - **Uncommitted data:** when one transaction is allowed to see intermediate results of another transaction before it has been committed
  - **Inconsistent retrieval:** when a transaction reads several values from the database but a second transaction updates some of them during execution of the first
- Concurrency control with locking methods
  - **lock**
    - guarantees exclusive use of a data item to a current transaction
    - required to prevent another transaction from reading incomplete data
  - **lock manager**
    - assigns and polices locks used by transactions
  - **lock granularity**
    - database-level lock: entire database is locked
    - table-level lock: entire table is locked
    - page-level lock: entire diskpage is locked
    - row-level lock: allows concurrent transactions to access different rows of the same table, even if rows are located on the same page
    - field-level lock: allows concurrent transaction to access the same row, as long as they require use of different fields/attributes within the row
  - **lock types**
    - binary lock: two states: locked (1) and unlocked (0)
    - exclusive lock
      - access is specifically reserved for the transaction that locked the object
      - must be used when the potential for conflict exists
    - shared lock: concurrent transactions are granted Read access on the basis of a common lock
- Three levels of database backup
  - Full backup or dump
    - complete backup; performed overnight or on weekends, requires that no users be connected

        - takes longest, requires most/ample storage
    - Differential backup
        - only the last modifications done to the database are copied (compare to previous full backup)
        - only backs up data that has been changed since last full backup and has not yet been backed up
        - can be done overnight
    - backup of transaction log
        - back up all transaction log operations that are not reflected in a previous backup copy
- **Deadlocks**
    - occurs when two transactions wait for each other to unlock data
    - possible only if one of the transactions want to obtain an exclusive lock on a data item
        - no deadlock can exist among shared locks
    - possible solutions
        - **prevention**: transaction requesting a new lock is aborted if the possibility is that a deadlock may occur
            - if transaction aborted, all changes made by transaction are rolled back and locks released
            - transaction is rescheduled
            - best when there is a high probability of deadlocks
        - **detection**: DBMS tests database for deadlocks
            - if a deadlock exists, one transaction is aborted, rolled back and rescheduled and other transaction continues
            - best when there is a low probability of deadlocks
        - **avoidance**
            - transaction must obtain all the locks it needs before it can be executed
            - avoid rollback of conflicting transactions by requiring locks be obtained in succession
            - however, serial lock assignment required increase response times
            - best when response time is not an important concern
- **ISSUES IN NETWORKING AND DATABASE SECURITY**
    - **database security**: protection of the database against un/intentional threats using (non)computer based controls
    - Security - applies to data, hardware, software, people, etc.
        - theft and fraud: does not necessarily alter data; opportunity should be reduced (conducted by persons)
        - loss of confidentiality: need to maintain secrecy over critical data, could lead to loss of competitiveness
        - loss of privacy: need to protect data about individuals
        - loss of integrity: results in invalid / corrupt data
        - loss of availability: when data or the system or both cannot be accessed

- **Threats** - a situation or event that will adversely affect a system and thusly an organization
  - loss of hardware, software, data, or maybe loss of credibility or client confidence
  - staff shortage, wiretapping, illegal entry (possibly by 4chan neckbeards), etc.
- **Counter-measures**
  - physical controls to administrative procedures: DBMS is only as good as its OS
  - Authorization: granting of a right or privilege, which enables a subject to have legit access to a system or system object
    - authentication of subjects requesting access
  - authentication: mechanism to determine whether a user is who they claim to be
  - integrity: prevent data from becoming invalid to prevent bad results
  - encryption: encoding of data that renders data unreadable without the decryption key
  - privileges (triggered): rights to access certain database objects
  - views (subschema): dynamic result of one or more relational operations operating on the base relation to produce another relation
    - virtual relation that does not actually exist, but is produced upon request
    - External view (what the user sees), internal view, something else (she said it was in classroom but i can't find it)
  - backup and recovery: periodically take a copy of the database and log file to offline storage
  - associated procedures
    - authorization/authentication, backup, recovery, audit, installation of new application software, installation/upgrading of systems software
- **Non-computer countermeasures**
  - security policy
    - what area of business is covered
    - responsibility of employees
    - disciplinary actions
    - required procedure
  - contingency plan
    - key personnel, contact info, technical requirements for operations transfer
  - personnel control: attitude and conduct of ppl
  - secure positioning and storage
  - escrow agreement: legal contract concerning software between devs and clients; a third party holds source code for client applications
  - maintenance agreement
  - physical access control: door entry systems, card readers, access to site, viruses,data protection

        - protection of personal data from unlawful acquisition, storage, and disclosure; provision of necessary safeguards to avoid destruction and corruption of legitimate data field
    - privacy laws: right of an individual not to have personal info collected, stored, and disclosed either willfully or indiscriminately

**Data Mining**

- Information collected in digital form
    - **Business Transactions**: time related, inter-business deals (purchases, exchanges, banking, stock); intra- (management of wares and assets)
    - **Scientific Data**: we have a lot of data, we get more new data faster than we can analyze
    - **Medical and Personal Data**: lots of data are peoples: manage human resources, understand market, assist clients, other stuff
    - **Surveillance Video and Pictures**: store video footage from video cameras
    - **Satellite sensing:** constant stream of data from satellites, more than people can deal with per second
    - **Games**: massive amount of statistics, ways to report and exploit the data
    - **Digital media**: more efficiently store audio and video, make it easier to broadcast
    - **CAD and Software engineering data**(I'm an engineer kappa): store and display massive amounts of computer aided design data, software engineering programs need powerful resources to run
    - **Virtual Worlds**: creation of virtual three dimensional spaces to store objects, managing data in these spaces
    - **Text reports and memos**: text communication is stored in digital form, create libraries from them
    - **The World Wide Web repositories**: bunch of different stuff on the Web connected with hyperlinks, dynamic and unstructured so it's redundant a lot, covers broad amount of topics
- **Data mining**: (aka Knowledge discovery in databases): non trivial extraction of implicit, previously unknown and potentially useful information from data in databases (data mining is actually part of the KDD)
- **Knowledge Discovery in Databases**:
    - **Data cleaning**: or data cleansing; phase where noise and irrelevant data are removed from the collection
    - **Data integration**: multiple data sources, often heterogeneous, may be combined in a common source
    - **Data selection**: data relevant to the analysis is decided on and retrieved from the data collection
    - **Data transformation**: data consolidation, phase which selected data is transformed into forms appropriate for the mining procedure
    - **Data mining**: clever techniques are applied to extract patterns potentially useful
    - **Pattern evaluation**: strictly interesting patterns representing knowledge are identified based on given measures

    - **Knowledge representation**: discovered knowledge is visually represented to the user; visualization techniques to help users understand and interpret the data mining results
    - steps are often combined: data cleaning/integration = pre-processing
    - KDD is iterative: evaluation measures can be enhanced, mining can be further refined,new data can be selected or further transformed, or new data sources can be integrated
- What data can be mined?
    - **Flat files**: most common data source; flat files are simple data files in text or binary format, data can be transactions, time-series, scientific measurements, etc.
    - **Relational Databases**: a relational database consists of a set of tables containing either values of entity attributes or values of attributes from entity relationships
        - tables have columns and rows; columns are attributes, rows are tuples; tuples is an object or relationship between objects, identified by a set of attribute values representing a unique key
        - data mining algorithms can take advantage of the structure; benefits from sql for data selection, transformation, consolidation; but also goes beyond with predicting, comparing, detecting deviations, etc.
    - **Data Warehouses**: repository of data collected from multiple data sources and intended to be used as a whole under the same unified schema
        - option to analyze data from different sources under the same roof
        - data warehouses are usually modeled by a multi-dimensional data structure
        - data cubes are well suited for fast interactive querying and analysis of data at different conceptual levels, known as Online Analytical Processing
        - Can be used in every situation, but emphasis on business decision making
            - Top, middle, and strategic management
            - Time dependant, changes in supply and demand
    - **Transaction Databases**:
        - set of records representing transactions, each with a timestamp, an identifier, and a set of items
        - transactions stored in flat files or two normalized transaction tables, one for transactions and one for transaction items
    - **Multimedia Database**: video, images, audio, text media; stored on object-relational, object-oriented databases or on a file system
        - high dimensionality, necessitates computer vision, computer graphics, etc.
    - **Spatial Databases**: store geographical information in addition to normal data
    - **Time-series**: time related data such as stock market data or logged activities
        - monitor continuous flow of new data, sometimes requires real time analysis

- **World Wide Web**: three components: content of the Web, structure of the web, and usage of the web

# MAY 2018 CASE STUDY: AUTONOMOUS TAXI

**Key Terms:**

1. **Autonomous:** Having the freedom to govern itself or control its own affairs.
2. **Backpropagation:** A common method of training a neural net in which the initial system output is compared to the desired output, and the system is adjusted until the difference between the two is minimized.
3. **BigO Notation:** Big O notation is the language we use for articulating how long an algorithm takes to run. It's how we compare the efficiency of different approaches to a problem.
4. **Bounding Boxes:** A bounding box (usually shortened to bbox) is an area defined by two longitudes and two latitudes, where- Latitude is a decimal number between -90.0 and 90.0.Longitude is a decimal number between -180.0 and 180.0.
5. **Brute-Force:** Brute force (also known as brute force cracking) is a trial and error method used by application programs to decode encrypted data such as passwords or Data Encryption Standard (DES) keys, through exhaustive effort (using brute force) rather than employing intellectual strategies.
6. **Convolutional Neural Networks (CNN):** In machine learning, a convolutional neural network (CNN, or ConvNet) is a class of deep, feedforward artificial neural network that have successfully been applied to analyzing visual imagery.
7. **Cost function:** used to chart how production expenses will change at different output levels.
8. **Deep learning:** Deep Learning is a subfield of machine learning concerned with algorithms inspired by the structure and function of the brain called artificial neural networks.
9. **Dijkstra's algorithm:** An algorithm to find the shortest paths from a single source vertex to all other vertices in a weighted, directed graph. All weights must be nonnegative.
10. **End-to-end learning:** End-to-end steering describes the driving-related AI task of producing a steering wheel value given an image.

11. **Features maps (Activation maps):** Feature Mapping is an interactive classification process that can be applied to any aerial or satellite multiband imagery, from high-quality hyperspectral to poor-quality air video.
12. **Filters (Kernels):** An image kernel is a small matrix used to apply effects like the ones you might find in Photoshop or Gimp, such as blurring, sharpening, outlining or embossing. They're also used in machine learning for 'feature extraction', a technique for determining the most important portions of an image. In this context the process is referred to more generally as "convolution".
13. **Filter stride:** Filter Stride-stride is the number of pixels with which we slide our filter, horizontally or vertically
14. **Greedy algorithm:** A greedy algorithm is a mathematical process that looks for simple, easy-to-implement solutions to complex, multi-step problems by deciding which next step will provide the most obvious benefit.
15. **Machine learning:** Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves.
16. **(max) pooling:**
17. **Multi-layer perceptron (MLP):** A multilayer perceptron (MLP) is a feedforward artificial neural network that generates a set of outputs from a set of inputs. An MLP is characterized by several layers of input nodes connected as a directed graph between the input and output layers. MLP uses backpropagation for training the network. MLP is a deep learning method.
18. **Nearest neighbour algorithm:** In pattern recognition, the k-Nearest Neighbors algorithm (or k-NN for short) is a non-parametric method used for classification and regression. In both cases, the input consists of the k closest training examples in the feature space.
19. **Overfitting:** A modeling error which occurs when a function is too closely fit to a limited set of data points. Overfitting the model generally takes the form of making an overly complex model to explain idiosyncrasies in the data under study. In reality, the data being studied often has some degree of error or random noise within it. Thus attempting to make the model conform too closely to slightly inaccurate data can infect the model with substantial errors and reduce its predictive power.
20. **Point clouds:** A point cloud is a collection of data points defined by a given coordinates system. In a 3D coordinates system, for example, a point cloud may define the shape of some real or created physical system. Point clouds are used to create 3D meshes and other models used in 3D modeling for various fields including medical imaging, architecture, 3D printing, manufacturing, 3D gaming and various virtual reality (VR) applications.
21. **Receptive Fields:** An area of the body surface over which a single sensory receptor, or its afferent nerve fiber, is capable of sensing stimuli. In some body area, e.g. face, ears, front paws, the sensitive areas are small; over the back they are larger.
22. **Sensor fusion:** Sensor fusion is the aggregation of data from multiple sensors to gain a more accurate picture of the sensors' subject or environment than can be determined by any one sensor alone.

23. **Society of Automotive Engineers:** professional association and standards developing organization for engineering professionals in various industries.
24. **Shift variance (spatial invariance):** A shift invariant system is the discrete equivalent of a time-invariant system, defined such that if y(n) is the response of the system to x(n), then y(n-k) is the response of the system to x(n-k).
25. **Vehicle-to-vehicle Protocol:** automobile technology designed to allow automobiles to "talk" to each other. V2V communications form a wireless ad hoc network on the roads. Such networks are also referred to as vehicular ad hoc networks, VANETs.
26. **Vehicle-to-infrastructure Protocol:** a communication model that allows vehicles to share information with the components that support a country's highway system. Such components include overhead RFID readers and cameras, traffic lights, lane markers, streetlights, signage and parking meters.