COMPUTER SCIENCE NOTES

SYSTEM FUNDAMENTALS

Planning and system installation

Reasons for a new system:

- Old system is inefficient
- Old system is no longer suitable for original purpose or outdated
- To increase productivity and quality of output
- To increase efficiency and minimise costs

Extent of a new system update depends on how much time, software, hardware, people needed and the immediate environment. May need to train employees to use the system, fire employees (e.g. secretary not needed if salespersons input orders from home PC), get more hardware (e.g. employees need PC and network needs to be implemented), change server location etc.

To decide whether project is worth pursuing (Compatibility issues/ strategies for mergers/ data migration/ hosting system/ installation processes are issues to be considered in the planning stage, once the project is deemed worth pursuing), use:

***T**echnical feasibility:* Is available technology sufficient and advanced enough to implement the system?
***E**conomic feasibility:* Is the new system cost effective? Are funds sufficient? ***L**egal feasibility:* Are there conflicts between the system and laws/ regulations? ***O**perational feasibility:* Are existing organisational procedures sufficient to support maintenance and operation of the new system?
***S**chedule feasibility:* How long will it take to implement?

Change management: Shifting people, departments and organisations from one state to the desired state. Need to maximise benefits and minimise impact of change on people so that stakeholders accept the change in environment. Issues regarding planning the system must be resolved, e.g. students informed of double-sided printing to make use of the new feature

Compatibility issues
Business mergers: Two businesses combine. Need to ensure systems are compatible. Incompatibility can arise from...

Language differences: Communication issues and different interpretations. ***Software incompatibility:*** Different software/ systems can't operate well on the same computer or same network.
Legacy systems: Old tech, hard/software, computer system or program. Some still satisfy user needs and data cannot be converted to newer formats, or applications upgraded, so organisation continues use of legacy systems even if newer technology is available.

Strategies for merging:

1. Keep both systems, develop to have same functionality (high maintenance cost)
2. Replace both systems with a new one (high initial cost)
3. Combine best systems from both companies (hard for employees to work with system from another company)
4. Only use one company's info systems (policy issues)

Other problems to overcome (expanded on in notes further below)…

- **Workforce issues-** Might have to lay off some workers, or retrain
- **Time frame-** to merge two systems
- **Testing-** of the combined systems/ new data
- **Costs-** of aligning two systems
- **Changeover decisions-** e.g. parallel running etc.

Using client hardware VS hosting systems remotely

Locally hosted system: Software is installed and operated on client's own hardware/ infrastructure. Like paying to buy the product/ software package and owning it. E.g. set up open source message board system on your own web server.

PROS:

- Best for large, complex systems.
- Only pay once, excluding maintenance (if you don't pay it, can still continue to use the software but not updated versions).
- Can control the data yourself in a secure data centre, less risk of data loss as you yourself can use redundancy to whatever extent you want.

CONS:

- Higher initial cost than remote
- Harder to predict total cost (maybe more expensive in long run with maintenance payments)
- Have to maintain yourself (hire IT personnel).

Remote hosted system: Software As A Service (SaaS) solution. Hardware is elsewhere, updated centrally. Users can access data and operate software from the cloud and pay for service on a subscription basis. e.g. sign up for message board system where others take care of maintenance.

PROS:

- Lower initial cost
- Can predict overall cost easier
- Best for when organisation doesn't have necessary hardware already
- You don't have to maintain it yourself
- Data secure in data centre

CONS:

- Relying on a third party= risk of data loss if provider shuts down
- Legislation in country of provider may be weaker than in user's country
- Performance generally lower than on-premise solutions
- Remote host in different time zone, can be inconvenient for end-users?
- Depends on internet connection

Installation processes (adv/dis)

Implementation/ conversion: Putting new system online and retiring old one. Types...

- **Parallel:** Both systems run parallel to each other at first to compare outputs until satisfied with the new system, terminate old one. If new system fails, can revert to old one = less risk, ideal for critical systems e.g. nuclear power station. But higher cost. Not efficient if systems have different input/outputs/ processes. Workers may be trained to use new system for nothing.
- **Direct/ Big Bang:** Set up new one and terminate old one at the same time. Preferred if system not critical, due to higher risk as system might not function well. Less costly.
- **Pilot:** In organisations with multiple sites. New system is introduced in one of the sites (pilot site/group) then introduced to others if successful. Less risk. Worker training
- **Phased:** Convert one module at a time e.g. per department. Training period and implementation takes longer

Problems with data migration

Data migration: Transferring data between formats, storage types and/or computer systems when switching to a new system/ changing/ upgrading/ merging. Problems/ risks...

- **Incompatibility** with file formats in the new system- could lead to incomplete or incorrect data transfer
- **Non-recognisable data structures** and formats- result in a mismatch of data, e.g. in customer records
- **Data lost or corrupted** during transfer due to transmission faults/lack of adequate storage. Not usable at destination
- **Data misinterpretation** due to conventions in different countries e.g. date, measurements, currencies
- **Different validation** rules between companies- could lead to inconsistent/incorrect results
- Might not be able to use data while transferring, problem if it's large and takes long

Types of testing

Testing is important because it identifies problems to be fixed, areas for improvement and determines whether system/ software fulfils requirements. If not done properly, inadequate system= inadequate employee productivity, reduced efficiency and output, increased costs= end user dissatisfaction.

1. **Alpha testing:** Offering early development version to other developers before available to general public, get feedback.
2. **Beta testing:** Provide version to select group of users outside of company (closed beta) or to public (open beta) and receive real-world feedback. But user report is not always best quality, and there are many reports of the same bugs.
3. **User acceptance testing:** Usually last stage, provided to clients as a last-minute check that the product satisfies target audience
4. **Debugging:** Systematically finding and correcting bugs/errors. Some programs do it automatically = cheaper and faster.

User focus

User documentation

Important so users can understand, use and make the most of the system. Ensures users can quickly adapt to the software/ system with minimal costs/inefficiencies. Can include...

Requirements- identify attributes, characteristics and functions

- **Technical-** details on how to install and configure the product

- **End user-** manuals for end user, support staff and system administrators. Details on how to use the product
- **Marketing-** how to market the product, analysis of market demand

Methods of documentation include...

Help files: Easy to access, cheap, can't be misplaced, no internet needed **Online documentation:** Easier to use and search through, can have email support and can update documentation later. However, access limited by internet connection **Manuals:** Printed manuals can be accessed any time (even if system not yet installed) and not restricted by internet connection but can't be updated. Digital manuals more cost efficient and eco-friendlier but online ones restricted by internet connection.

User training methods (adv/dis)

- **Self-instruction:** People can use resources like manuals, websites, video tutorials etc. to learn on their own. Easiest and cheapest with more flexible time for user but usually only used for easy/ common use programs with sufficient documentation as effectiveness depends on user motivation and ability to work on their own.
- **Formal classes:** Classroom setting, free discussion. Students can exchange ideas, direct interaction with expert. But may be harder for members who work better on their own and self-assured students may dominate discussions
- **Remote/ online training:** Larger variety of courses online, can access any time, easier to set up and include new members = cheaper. But excludes those without infrastructure/ internet or IT skills to use it, might not be as effective (especially with dependent learners)

In general, employees need to learn quickly and easily to implement new system faster to reduce costs and minimise inefficiencies

System backup

Causes of data loss:

- **User error:** Accidental deletion, closing before saving
- **Natural disasters:** Fire, flood, earthquake
- **Malicious activities:** Someone purposefully deleting/ altering/ stealing data (can be employee or external hacking)
- **Computer viruses:** Destruction/ corruption of data
- **Power failure**

Consequences can be serious e.g. hospital: puts lives in danger, may have to repeat tests and procedures. In other situations, can cut into revenue if dissatisfied customer tells others e.g. customer makes reservation but there's no record of it or free rooms so they have to go elsewhere.

Measures to prevent data loss

1. **Regular backups:** On hard disks/ magnetic tape, online or on ***removable media*** (e.g. USB, CDs) for fast backup and storage
2. **Offsite storage:** Data backups stored in different geographical location
3. **Firewall and antivirus:** Prevent virus infections
4. **Failover systems:** Computer system that system can switch to in case of hardware/

software/ network failure. Often switches automatically to reduce time

Software deployment

It's important that users can install updates because otherwise they might not have fixes for bugs and errors or be able to benefit from added features/ improvements leading to performance issues. Especially for organisations with different locations- different sites could have different versions of the software, leading to incompatibility.

Types of updates (i.e. reasons for updates)

- **Patches:** Used to fix known bugs and vulnerabilities. May introduce new bugs though
- **Updates:** Fix bugs, add minor functionalities, usually free
- **Upgrades:** As well as bug fixes, add new major functionalities or characteristics, often need to be bought
- **Releases:** Final, working applications gone through testing

Strategies for alerting users about updates

- **Automatic:** Cookie is placed on the machine when software is registered and installed, communicates with the developer automatically when software is started up. If update is available, messages and alerts are sent back to the machine
- **Sending an email:** User registers email and other details when installing software. Email sent to the registered user with a link to download the update

Topic 1.2

Computer components

Hardware: Physical, tangible elements of a computer system e.g. CPU, HDD ***Software:*** Set of instructions for the CPU to perform specific operations, can be programs or data ***Peripheral device:*** Auxiliary device that can connect to, communicate and work with the computer. Input/ output devices.
Human resources: The set of individuals who make up the workforce of an organisation, business sector or economy. There is software that combines human resources functions (e.g. payroll, recruiting and training, performance analysis) into one package

Roles of a computer in a networked world

- **Client:** Piece of computer hardware/software that accesses services made available by server, by sending requests to server
- **Server:** Program/host computer that fulfils requests from client programs or computers across network and shares info to clients
- **Email server:** Message transfer agent that transfers electronic messages from one computer to another in a network
- **DNS (Domain Name Server):** Server that translates web addresses written in letters (more memorable for humans) to the numeric IP (Internet Protocol) address
- **Router:** Connects networks together to forward data packets between networks, deciding where to send information so it is received by one network and then sent to another until it reaches its destination

- **Firewall:** Controls incoming and outgoing network traffic, determining what data packets should be allowed through, based on a rule set. Needed to protect integrity of client computer.

Ethical/ social issues with the networked world (interconnecting computers):
(also ethical issues associated with introduction of new IT systems)

- **Security:** Protecting hardware, software, peripherals, data and networks from unauthorized access
- **Privacy:** Controlling how and to what extent data is accessed and used by others, to protect identity e.g. GPS location services on phone, data sold to companies. But there's also problems with anonymity e.g. cyber bullying, hacking, terrorism etc.
- **Censorship:** Some info may be deemed inappropriate. Network manager could make sure no other computers can access it. e.g. China blocking sites.
- **People and machines:** Easier communication, more information and efficiency etc. BUT addiction, real life neglect, lack of sleep, health problems, car accidents, technical unemployment, digital alterations (e.g. fake videos, fake news etc).
- **Digital divide and equality of access:** Inequalities regarding use and access to computer systems in different environments/ countries, leads to inequality in info and education access
- **Surveillance:** Monitoring people e.g. for law enforcement, employers, traffic control etc. Ethics of privacy and knowledge/ consent to surveillance
- **Globalisation & cultural diversity:** Spread info and reduce political, geographical, financial boundaries. BUT diminishing of traditional cultures
- **Environment**
- **How the system will benefit the company**

Practical issues to consider when networking:

- **Reliability:** How consistently a computer system functions according to its specifications, with minimal system failure. Having a long mean time between failures. Failure can = data/time/revenue loss, injury etc.
- **(Data) integrity & consistency:** Maintenance of accuracy and consistency of data. Must be complete, up-to-date, unaltered. Is inconsistent if there's different versions of data (duplication)
- **Standards and protocols:** Rules followed in development of systems, including proprietary standards (e.g. computers compatible with Microsoft operating system), industry standards (formally decided, e.g. USB), de facto standards (e.g. QWERTY keyboard)

System design and analysis

Stakeholder: Has an interest or investment in a project and is impacted by how it turns out. System analysts have to collaborate with all stakeholders- clients and end-users.

Obtaining requirements:

- **Interviews:** Face-to-face with verbal responses. Can be structured with the same questions and manner for every stakeholder, or unstructured with more flexibility. - **Questionnaires:** Can be closed/restricted (yes/no, box checking) or open/unrestricted (free response questions)
- **Direct observation of current procedures:** On-site observation of different departments to see where things can be more efficient.

Evaluating requirements

Interviews:

- Talk directly to users/member of organisation and can observe non-verbal behaviour= more reliable, valid data
- Unstructured interviews can reveal more questions that otherwise wouldn't have been addressed= more detailed reports
- Data from unstructured interviews is hard to summarise/ evaluate/ analyse Level of detail depends on type of interview- structured interviews get less detailed responses
- Time-consuming to get detailed results

Questionnaire:

- Time-saving and cost-efficient- can get info from a large group of people easily and cheaply
- Closed/restricted questions= data is easier to compare
- Open questions= more detailed reports
- Level of detail depends on type of questions- closed questions don't allow for clarifications, elaborations, more details
- Stakeholder could interpret question wrong= invalid answers

Observations:

- Highlight aspects not detected in questionnaires/interviews= produce more detailed reports
- May be more reliable than interviews- see what people actually do, instead of what they say they do
- Time consuming/expensive- might need to observe a complete business/system cycle which could take a significant amount of time
- People act differently when they know they are being watched= unreliable observations

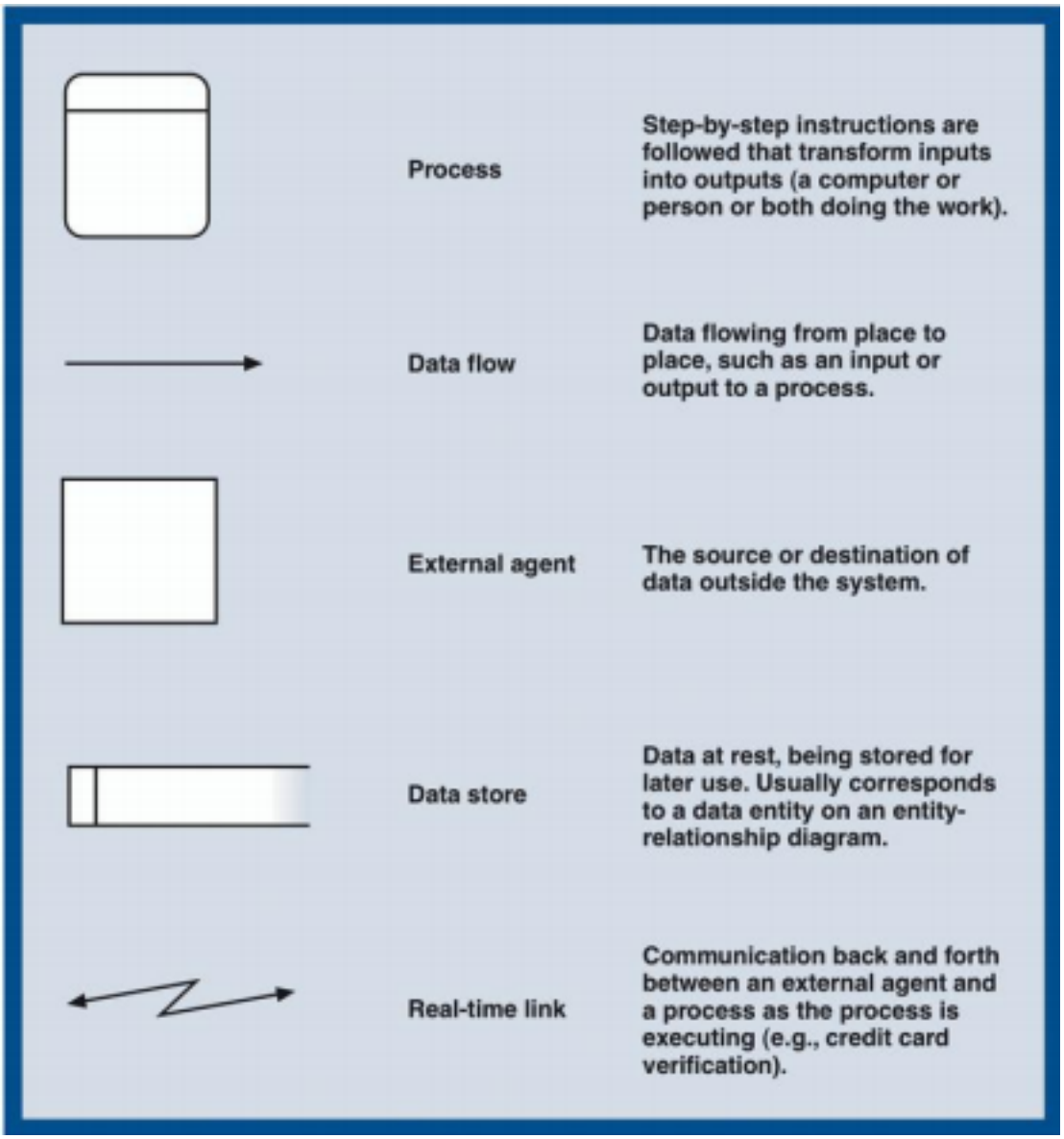
Prototype: Early sample, version or model of a system/software/hardware, displaying the minimum necessary features, used to test and gather feedback on a new concept or system from clients. Clients can follow development closely and see the changes as they are made. ***Iteration (iterative design):*** Where solutions/code/prototypes are designed, developed, tested and evaluated in repeated cycles. With each iteration, additional features may be added until there is a fully functional software.

This involves end user participation. Failure to involve end user in design process can lead to software not suitable for its intended use because of lack of feedback- has adverse effect on user productivity, efficiency etc. Need effective collaboration and communication between client, developer and end-user.

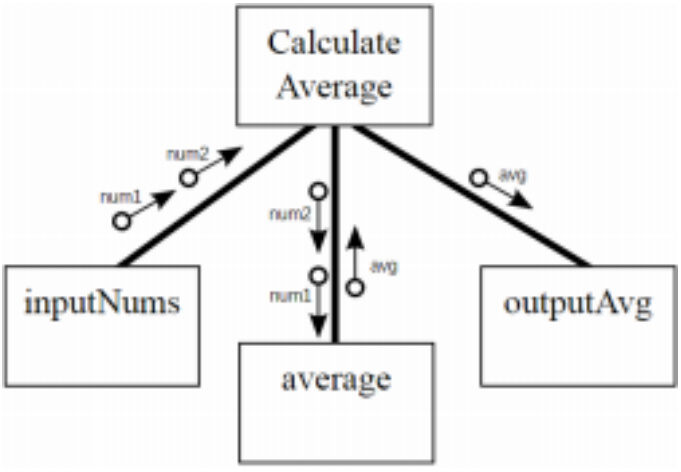
Illustrating system requirements:

Flowcharts show flow of data through program, can show all types of processing and can refer to hardware as well as programs, files, databases etc.

Data Flow Diagrams (DFD) show how data is stored and moved through the system, but not type of data or storage.



Structure charts describe functions and sub-functions and relationships between modules in a program. Involves modular design, process of designing modules individually then combining to form solution.



Human interaction with the system

Usability: Ability to accomplish user goals. More usable= more efficient to use, easier to learn.
Accessibility: Ability of system/ device to meet needs of as many individuals as possible. Low

accessibility = barriers to certain groups e.g. disabled

Usability problems and examples:

- **Learnability:** *How easy is it to accomplish basic tasks the first time users encounter the design?* E.g. learning features of different manufacturers, accidental touches on touch screen, right-handed mouse
- **Efficiency:** *How quickly can users perform tasks?* E.g. need to locate product and details quickly on e-commerce sites
- **Memorability:** *When returning to design after period of not using it, how easily can users establish proficiency?*
- **Error:** *How many errors do users make, how severe, how easily can they recover?* e.g. inaccurate/ outdated street data, no verification/ validation, time taken to reschedule
- **Satisfaction:** *How pleasant is it to use the design?* E.g. visually appealing website, all other factors carried out

Can be affected by...

- **Complexity/ simplicity:** e.g. unnecessary extra apps when user just needs call & SMS, website clearly stating what company offers and clear navigation, unclear instructions
- **Readability/ comprehensibility:** e.g. small screen/ buttons, low quality speakers, incomprehensible font and colours
- **Other stuff:** Battery life, brightness outside, health problems, hardware components e.g. camera with no flash, insufficient memory

Example- *usability issues with mobile devices…*

- Size of screen- difficult to see/use especially in poor light
- Size of keys- difficult to access functions
- Battery life- may need to recharge regularly, inconvenient

Solving accessibility/ usability problems:

Usually for disability or impairment

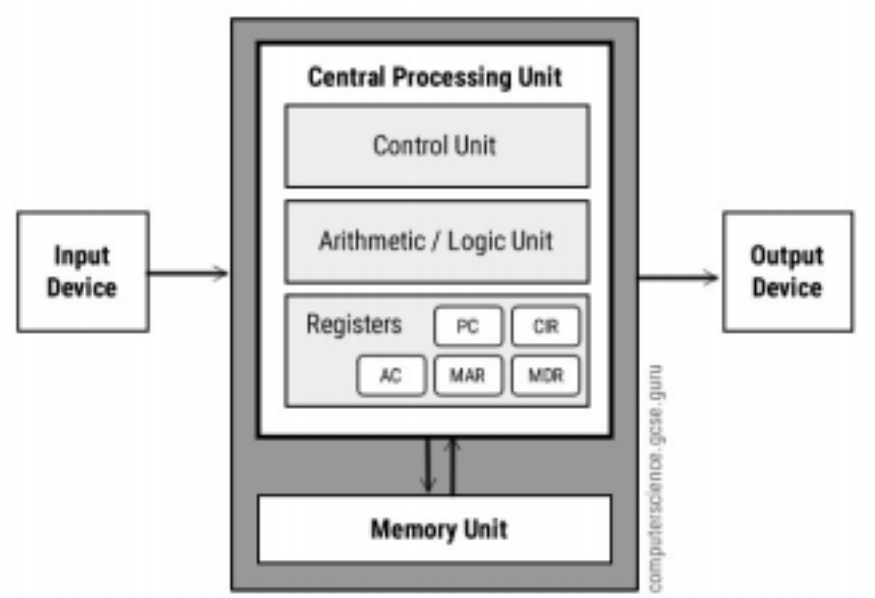
- **Visual impairment:** Braille keyboard/ printer, speech recognition, text-to-speech/ screen readers, colour changers, screen magnifiers
- **Hearing and speech:** Subtitles, visual effects
- **Cognitive problems:** Word processors for dyslexics, special software with strong interaction
- **Mobility issues:** Eye typer, puff and suck switch, foot mouse, speech recognition, word prediction software

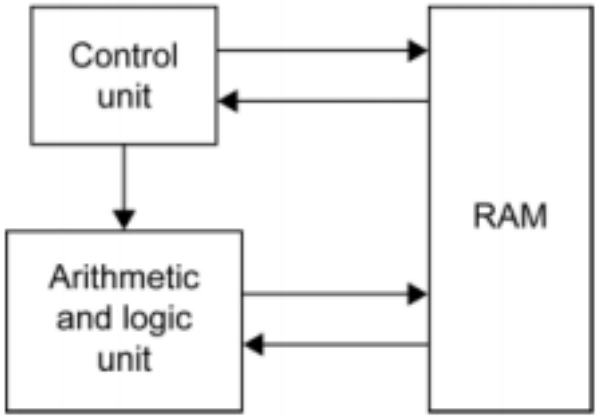
Topic 2

COMPUTER ORGANISATION

Computer architecture

Central Processing Unit (CPU): Component of a computer system which interprets/ processes and executes program instructions.





<- From markscheme

Control Unit (CU): Fetches data/ instructions from memory, decodes it into commands and controls transfer of data/instructions among other units of CPU.
Arithmetic Logic Unit (ALU): Performs arithmetic and logical calculations/operations

Registers: Small, very fast circuits that store intermediate values from calculations or instructions inside the CPU

- ***Memory Address Register (MAR):*** Stores address of next instruction to be decoded and executed. Connected to RAM through Address Bus
- ***Memory Data Register (MDR):*** Stores data of instruction most recently taken from RAM, i.e. instruction about to be executed. Connected to RAM through Data Bus

Instruction/Machine cycle

Basic operation cycle of computer taking place within certain time period, where instruction is fetched, decoded, executed and stored.

1. Address of next instruction stored in MAR (through Address Bus)
2. Using this address, data is fetched from main memory and stored in MDR
3. CU decodes instruction
4. ALU executes command/ operation
5. Output value stored in accumulator if needed
6. Processed data stored back on main memory. Position/address of location the data will be stored in is sent along Address Bus, with data sent along Data Bus

Bus: Set of wires that connect two components in a computer system

Primary memory: Only storage directly accessible by CPU

Random Access Memory (RAM): Stores data and instructions of program currently being run. Data stored in memory locations that each have a unique address that the CPU uses. Data constantly moved and overwritten. RAM is volatile- all data/ instructions are deleted once there's no power

Read Only Memory (ROM): Stores permanent instructions and data of programs used to boot and operate the computer (e.g. Basic Input Output System). Cannot be overwritten. Non-volatile, data and instructions remain even when there's no power. Smaller than RAM.

Cache memory: Type of memory that is smaller and accessed faster than RAM since closer to the CPU. Stores most frequently used data and instructions so CPU has to access the slower RAM less often and performance speeds up. Computer checks if there's a copy of data in cache before accessing main memory.

Secondary memory

Examples: HDD, USB, SD, magnetic tape, floppy disk

Or auxiliary storage. Relatively slow but higher capacity than primary memory. Can be written to but non-volatile. Known as ***persistent storage*** because contents are persistent, not wiped out after process is finished, or when there's no power.

Data and instructions are copied from secondary storage to main memory (RAM) before being executed by CPU. Because RAM has limited capacity & processed data needs to be stored somewhere non-volatile, processed data and instructions are moved to secondary memory to make space for new ones.

Without persistent storage, computer would only be able to perform basic operations and all data re-inputted every time. Need to store processed data so we can use it again later, e.g. editing a saved file.

Virtual memory: Part of secondary memory used as if it is main memory. If RAM is too full to load new program, another program/ part of another program is loaded on virtual memory to make room for the new program. If the other program is needed again, it's swapped back out

Operating and application systems

Operating system: Set of software that controls computer's hardware resources and provides services for programs.

Roles of an operating system:

1. **Peripheral communication:** OS controls hardware and provides interface for applications to communicate with peripherals
2. **Memory management:** Manages how memory is used by applications. OS allocates sections of memory to each program being run, keeps track of what data is in which location. If there's too many processes for RAM, uses virtual memory to run programs simultaneously
3. **Data management:** Accessing and storing data. Keeping track of files and location in memory to make best use of memory available and provide reliability
4. **Resource allocation:** Manages resources and multitasking so tasks share CPU time, including memory and processor time
5. **Security:** Provides measures such as password authentication, magnetic cards, access rights to prevent unauthorised access. Log files keep track of activity

Application software:

- **Word processing:** e.g. MS Word, Google Docs. Producing, editing, formatting and printing of documents.
- **Spreadsheets:** e.g. MS Excel, Open Office Calc. Data represented by cells, organised in rows and columns. Can perform calculations through formulae to process data and present it in visual charts to analyse.
- **Database Management System (DBMS):** e.g. MS Access. Manages and provides interface for users to use database (organised store of data). Data is organised in records, user can create or modify records, query and extract data.
- **Web browsers:** e.g. Chrome, IE, MF. Access, retrieve, present content on WWW. Connects to web servers to request information.
- **Email:** e.g. Gmail (web-based) or MS Outlook (software-based). Exchange of digital messages to email server, recipient receives message when they connect to server
- **Computer Aided Design (CAD):** e.g. Autodesk AutoCAD, Google Sketchup, Dassault Systemes Solid Works. Often used in engineering, manufacturing and architecture to create, modify and analyse a design. Can convey info like shape, materials, dimensions etc. with changeable values
- **Graphic processing:** e.g Adobe Photoshop, Adobe Illustrator, GIMP. Manipulate visual images: move, erase, crop, colour etc.

Common features of applications:

Graphical User Interface (GUI) allows user to interact with software in different ways. Replaced ***Command Line Interface (CLI)*** where you had to type in commands. GUI includes...

- **Toolbar:** Buttons, icons, menus etc. on it
- **Menu:** List of commands you can chose
- **Dialogue box:** Communicates info to user and allows to choose option

These are common elements provided by OS to improve usability as they're usually in the same place and used in the same way, while others may be provided by the software. Use of GUI improves usability.

Binary representation

Binary digit (bit): Basic unit of info, 1 or 0. One digit in a binary number.
Byte: 8 bits. Kilobyte: 1024 bytes. Megabyte: 1024 kilobytes.
Decimal/ denary: Positional system that uses 10 digits to represent a number. Base-10.
Binary: Uses 2 digits to represent a number. Base-2
Hexadecimal: Positional system that uses 16 digits to represent a number. Base-16
0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

1011011_2 = base 2, therefore binary
104_{10} = base 10, therefore decimal
84_{16} = base 16, therefore hexadecimal

Converting

128 64 32 16 8 4 2 1 1 0 1 1 1 0
^Binary representation of [00]101110. Add up all the values where there's a 1. Converting from Dec to Binary could involve algorithm: divide by 2, retain remainders until you're dividing 0 by 2. Rest of the digits are 0 if you haven't reached 8 digits. Read result last to first to get binary number left to right.

e.g. 104/2 = 52 remainder 0
52/2 = 26 remainder 0
26/2 = 13 remainder 0
13/2 = 6 remainder 1
6/2=3 remainder 0
3/2 = 1 remainder 1
1/2 = 0 remainder 1
0 (add a zero to make 8 digits) 104= 01101000

16^3 16^2 16^1 16^0 4096 256 16 1 1 B 5
^Hex representation of 1B5. Decimal value = (1*256)+(11*16)+(5*1) = 437. Or algorithm: keep dividing by 16 and keep remainder, then put them together (write letters for digits over 9). Put first value in right-most box.

Data representation:

- **Integers:** 256 values can be represented by 8-bit numbers. If representing negative numbers most significant bit (MSB, most left one) is either 0 (positive) or 1 (negative) so only 7 bits are left for the number.
- **Characters:** American Standard Code for Information Interchange (ASCII) character encoding scheme represents text. Each character (low/upp case latin letters, numbers, punctuation and control characters) is assigned a value from 0 to 127. Unicode can also have symbols and other languages

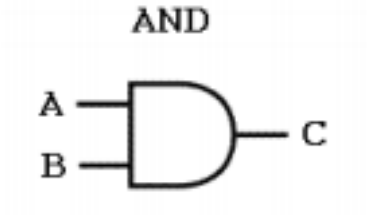
Strings: e.g. UTF-8 (unicode) used. On average 40 bits per word needed **Colours:** Pixels, each one has a colour. Colour value can be represented by hex RGB- 6- digit hex number (3 values), every 2 digits show how much red/green/blue there is in the colour. Can

also be represented in binary; more bits = more colours. Resolution = width*height in pixels.

Logic gates

AND

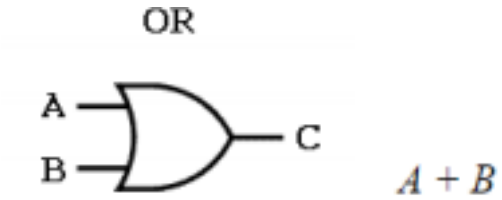
2 inputs, output TRUE only if both inputs TRUE



A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

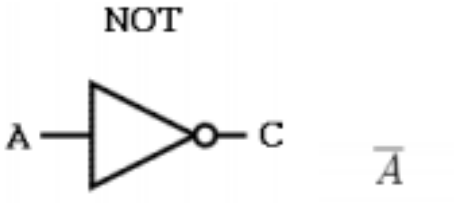
OR

2 inputs. If either input is true, output is true.



A	B	C
0	0	0
0	1	1
1	0	1
1	1	1

NOT

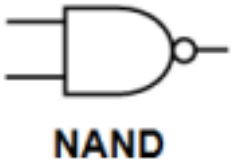


1 input. Output is inverse of input.

A	C
0	1
1	0

NAND

NOT AND. AND gate but the output is inverted, so when both inputs are TRUE, output is



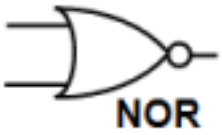
$\overline{A \cdot B}$

FALSE.

A	B	C
0	0	1
0	1	1
1	0	1
1	1	0

NOR

NOT OR. OR gate but output inverted, so opposite of OR gate. If either input is true, output is



$\overline{A + B}$

false

A	B	C
0	0	1
0	1	0
1	0	0
1	1	0

XOR (exclusive OR)

Like OR- if either input true, output = TRUE. Except for when both inputs = TRUE, then output = FALSE.

$A \oplus B$

Or:

$A \cdot \overline{B} + \overline{A} \cdot B$

A	B	C
0	0	0
0	1	1
1	0	1
1	1	0

Logic gate diagrams in IB only need circle with name inside

Constructing truth tables

Example: Tania won't go to play if it is cold and raining, or if she has not done her homework.

Won't play = (Cold AND Raining) OR (NOT Homework)

$\overline{W} = (C \bullet R) + H$

C	R	H	C • R	$\overline{\ }$

W

H

0 0 0 0 1 1 0 0 1 0 0 0 0 1 0 0 1 1 0 1 1 0 0 0 1 0 0 0 1 1 1 0 1 0 0 0 1 1 0 1 1
1 1 1 1 1 1 0 1

^Truth table for the 3 inputs, then figure out (C AND R) and (NOT H) for each combination of inputs between C, R and H. Finally figure out W for each combination by doing NOT of (C AND R) and (NOT H)

NETWORKS

Network fundamentals

Network: Set of computers or computer systems that are interconnected and share resources and data

Node: Device on network
Server: Computer system or software package that provides service to other devices in network, e.g. ability to file share. Fulfils requests from client programs or computers. ***Client:*** Device that requests service from server in the same network, e.g. email software client requesting email server software to fetch new emails
Hub: Connection point for all devices on network, devices connected through Ethernet cables. Data from device goes to the hub, then sent to all devices. Target device receives, others ignore
Switch: Connection point but can identify which device is connected to which port, so can send data to target device
Router: Device that connects network to another e.g. home network to internet.

Types of networks

- **Local Area Network (LAN):** Network devices are connected within limited geographical area, e.g. a room, a home, a school, office building etc. Data and peripherals can be shared. Usually client-server, connected with hub/ switch and Ethernet cabling
- **Virtual Local Area Network (VLAN):** Devices or LANs from different segments in a LAN connected logically, e.g. splitting a hospital LAN into VLANs according to department so each department can only access certain data. Usually via software, not hardware
- **Wireless Local Area Network (WLAN):** Connects nodes in limited geographical area but with no wires. Allows for mobile devices and easier connection/ removal, although less secure
- **Wide Area Network (WAN):** Connects different computer systems or LANs from different geographical areas, can span over a city, country or the world, e.g. internet, different sites of organisations connected
- **Storage Area Network (SAN):** Type of LAN designed to handle large data transfers. Each client is connected to a server that provides access to storage devices
- **Personal Area Network (PAN):** Network covering individual's working area, very small space. e.g. PC, phone, tablet sharing data. Can be wired (e.g. USB) or not (e.g. Bluetooth)
- **Peer-to-Peer Network (P2P):** Different to client-server, no centralised server. All nodes have equal privileges and responsibility for processing and sending data. All are client and server at same time, consuming and supplying resources from and to other peers

Internet: A WAN. Connect to it through internet service provider (ISP) for monthly fee. Not owned by single entity, no central server. All computer systems are independent to share resources. Internet provides services such as support for email, file transfer, and the WWW, which consists of webpages

Extranet: External extension to a LAN- Part of a network that uses internet protocols to allow controlled access by specific users to LAN or WAN. e.g. business wants to share some data/ information with clients or partners but not all, so extends part of network (extranet) available for access but with security/ privacy measures (e.g. firewall) so outsiders cannot access secure data

Virtual Private Network (VPN)

Allows clients from distant locations to connect, that otherwise wouldn't be able to connect with LAN (too far for cables) or WAN (too far for signal to be picked up). e.g. if worker goes abroad but still needs access to the network.

Technologies needed...

- Can use hardware for public networks like the internet through ***tunnelling***, which allows the network to send data via other network's connections as if connected to LAN
- Hardware/ software requirements like internet access, VPN software, routers
- Protocols to encrypt and authenticate traffic to ensure secure transfer over virtual tunnel, e.g. IPSec (Internet Protocol Security)

Advantages:

- Info can be accessed in remote places
- No need for long-distance leased lines = lower cost
- Enhanced security through encryption, can hide your IP address

Disadvantages:

- Needs a professional with detailed understanding of security issues and configuration to ensure sufficient security and protection
- Reliability of VPN is not directly under the organisation's control, but under ISP
- Not all VPN products are compatible across different vendors

VPN can change working patterns, e.g. changes in costs, flexibility to employees and remote offices, global networking opportunities, less time travelling for remote users

VPN vs Extranet

- VPN authenticates sender before establishing tunnel
- VPN access and transmission are always encrypted, whereas extranet has limited encryption
- VPN users have access to everything whereas extranet users only have access to specific services

Importance of standards in construction of networks

We need common rules/ standards when forming a network, otherwise some computer systems may not be able to connect due to incompatibilities, especially when trying to connect internationally. e.g. if there's no standards, could have a system developed to only support USB and a switch developed that only has Ethernet ports.

Network communication layers

An application goes through different layers to send data between systems. Data packets go through different layers with different protocols before being reassembled at the other application.

Open Systems Interconnection (OSI) model: Model that defines layers of network interaction. Each layer deals with one aspect/ abstraction of network communication

1. **Physical-** e.g. cabling system components
2. **Data link-** e.g. Network Interface Card (NIC)
3. **Network-** Routing
4. **Transport-** Transmission- error detection
5. **Session-** Retransmission of data if not received by device
6. **Presentation-** Encryption and decryption of message for security
7. **Application-** The end-receiver application, e.g. e-mail

Layers 1 – 3 = mostly physical communication
Layers 4 – 7 = mostly virtual communication

Acronyms to remember it: Programmers Do Not Throw Stale Pizza Away, Please Do Not Teach Students Pointless Acronyms

Data transmission

Data packet: Unit of data for transmission with a format, it is part of a message made into a single package. Contains address and data.
Packet switching: Network communication method: routers direct data packets transmitted along different paths through networks to the best next step closer to destination address (where there's not too much traffic, or no down devices blocking transmission). Eventually the packet reaches the router that knows the address of the destination device. Packets assemble there

Protocol: Sets of rules to facilitate a process being carried out correctly.
In case of data transfer, protocols are rules that ensure data is transferred correctly between systems. A protocol recognised as the standard for a certain type of transfer is called ***standard protocol***.
Protocols are concerned with…

- Ensuring the presence of an identified sender and receiver
- Ensuring the presence of an agreed method of communication
- Rules about format
- Data compression
- Type of error checking
- Recovery of data (Session layer)

Examples...

- Hypertext Transfer Protocol (HTTPS)- creates secure transmission of data from client to server
- Secure Socket Layer (SSL) and Transport Layer Security (TLS)- encryption protocols used on the internet
- Internet Protocol Security (IPSec)- encrypt and authenticate traffic to ensure secure transfer over VPN tunnel
- Dynamic Host Configuration Protocol (DHCP)- Allows server to automatically assign IP

address to client device
- Wireless Application Protocol (WAP)- For accessing info over mobile wireless network
- In transport layer, define the methods for opening and closing communication

Necessity of protocols

- **Data integrity:** Ensures data is not changed or corrupted during transmission **Flow control:** Controlling flow/load of resources due to limited bandwidth. Transport layer uses protocols, e.g. to request sending application to slow down data flow rate **Deadlock prevention:** Prevents situation where two or more competing programs or actions are sharing a resource and preventing each other from using it
- **Congestion management:** Prevents requests on network resources exceeding capacity
- **Error checking**

Speed of data transmission

Bandwidth: Amount of data that can be transmitted over a certain period of time, i.e. theoretical rate of data transfer or speed. Measured in bps or kbps
Throughput: Actual transfer rate of data
Bottleneck: Created where there are slower segments in a network due to different factors below. Throughput becomes only as fast as the slowest link or path

Factors affecting speed of data transmission:

- Different parts of a network use different media- e.g. fibre optics are faster than coax cable
- Bandwidth of network
- Data transfer rate and read speed of storage devices
- No. of connected devices can increase amount of traffic (demand and requests made), leading to congestion
- Time required for security and authentication checks
- Packet loss and retransmission- delayed by congestion

Data compression

Data is often compressed (encoding data using fewer bits than the original) to take up less bandwidth and reduce transmission time. Receiver has a program to decompress the file

Lossy compression (i.e. problems with compression): Permanently deletes certain information, only part of the original data is displayed when decompressed. Acceptable with formats such JPEG, GIF, MP3 as you won't notice much of a change although there may be less quality, e.g. smaller range of colours
Lossless compression: Only eliminates statistical redundancy, all original data can be seen when decompressed, e.g. Spreadsheets, word files, financial data as missing words or statistical data would be a noticeable problem

Transmission media

- **Wireless:** e.g. microwave/ radio signals, satellites, infrared (short distance). Compared to metal cabling and fibre optics, is least reliable and slowest but cheapest
- **Metal conductor:** e.g. copper cable, UTP cable, Coaxial cable. Faster, more reliable and expensive than wireless but less than fibre optic.
- **Fibre optics:** Fine optical fibres carrying beams of light as signals. Fastest and most

secure but also most expensive.

Wireless networking

Wireless devices communicate with no cables, e.g. with radio signals. Cell phones, tablets, LANs etc.

Advantages:

- Easier and cheaper to install than cable
- Easier to set up temporary links
- Easier to add new devices to network
- Many public access points exist, can work from different areas like home
- Can connect mobile devices, which are more convenient and portable

Disadvantages:

- Relatively slow communication as signals travel through air
- Affected by weather
- Weaker security and protection
- Higher error rates
- Health concerns about long term exposure to electromagnetic waves

Hardware and software components

- **Modem:** Allows you to connect to internet
- **Wireless router:** Same as router but also has ***Wireless Access Point (WAP)*** to connect to existing wired network and allow wireless data transfer between device and network
- **Wireless repeater:** Receives signal from router or access point and rebroadcasts it
- **Dynamic Host Configuration Protocol (DHCP):** Allows server to automatically assign IP address to client device.
- **Wireless Application Protocol (WAP):** Provides protocols for accessing info over a mobile wireless network, used in mobile devices to access internet, emails etc.
- **Firewall:** Controls what data comes in and out of network
- **SSID:** Service Set Identification. Set of 32 characters to differentiate WLAN from another
- **NIC drivers:** Operates hardware of the card. Translator between card and applications/ operating system of device
- **Web browser:** Software to receive and display content on WWW

Characteristics of wireless networks

- **Wi-Fi:** Short range wireless network offering high bandwidth for data transfer
- **1G/ 2G:** First/ second generation of mobile networking and telecommunications technology. Used analogue transmission
- **3G/ 4G:** Third/fourth generation “. Digital, faster than 1G/ 2G
- **WiMAX:** Worldwide Interoperability for Microwave Access. Serves long-range networking, even in remote areas
- **Future networks:** 5G or tactile internet

Lowest rate of data transfer- 1G/2G
3G

WiMAX (cheaper than 4G though)
4G

Highest rate- 5G (predicted)

WiFi varies between cards, maximums range from 11Mbps to 1,300Mbps. Faster than 3G/4G in short distance but less coverage

Network security measures

- **Encryption:** Altering a message into form not understandable to unauthorised person. Only person with the correct key can decode the message and read it.

Symmetric-key encryption: Or single/secret key. Device has a key to encrypt packets, the receiving device has the same key to decrypt. E.g. Data Encryption Standard (DES)

Public-key encryption: Or asymmetric key. Has a public key for encryption and a private one for decryption, both mathematically linked. Secure Socket Layer (SSL) and Transport Layer Security (TLS) encryption protocols used on internet

- **MAC addresses:** Medium Access Control. Every device in a network (their NIC) has a unique MAC address of 12 characters. Network permits access only to specified, trusted devices based on their MAC address. MAC address of an untrusted is device blocked so it cannot connect to the network.
- **User ID:** Use password to access device and/or password to access web interface used to set up wireless routers or access points
- **Firewall:** Either software installed on each device, or router hardware firewall that protects from hackers accessing devices through wireless connection. Filters incoming traffic and can block some messages coming through, as well as control/ limit user's own access to the internet. Extent of firewall use depends on an organisation's policy, e.g. one organisation may only allow communication between users and outside the network via email but not accessing websites etc.

Evaluating security methods

Symmetric-key encryption

- Faster than public-key
- Uses less computer resources
- Keys must be shared before used
- Danger of key becoming known by unauthorised individuals, another one must be used

Public-key encryption

- Two sides don't need to share secret key beforehand to communicate
- Messages take longer to encrypt and decrypt
- Authenticity of public key needs to be verified

MAC addresses

- Extra security
- Danger of allowed list of MAC addresses being discovered
- Difficult to manage the list as it grows

User ID

- Easy to use
- Prevents unauthorised access
- With web interface password, person cannot access web-based utility page of router/

modem/ access point unless they have the password
- Entering password each time may be time-consuming
- Weak password is easy to crack

Firewall
- Software firewall monitors traffic between device and network and prevents unauthorised access
- May slow down the device
- Issues about censorship with software firewall, depending on organisation
- Router hardware firewall prevents unauthorised persons getting access to the network
- Router hardware firewall can't stop person within range of your Wi-Fi getting onto your Wi Fi network

COMPUTATIONAL THINKING

Thinking procedurally

This includes identifying the steps and putting them in the correct order e.g. recipes

Sub-procedure: A section of code in a program that does a specific job. Can be called by name when needed without naming the details as these are wrapped in the procedure. It is therefore an example of abstraction.

Thinking logically

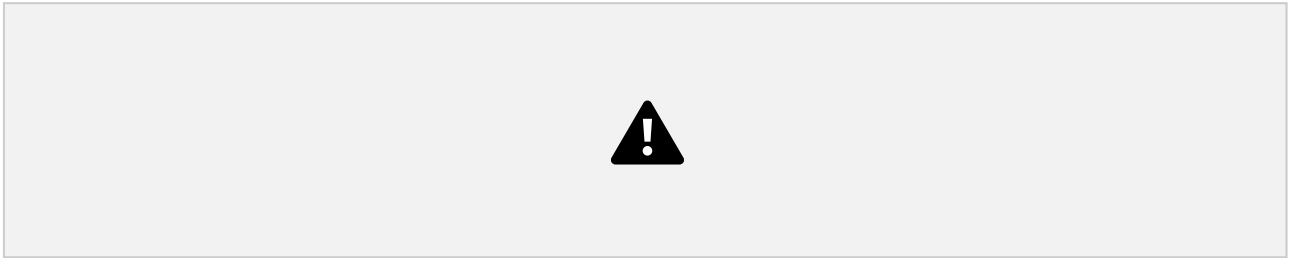
Different actions are taken based on conditions, taking alternative procedures into account. Need to identify conditions associated with a given decision (like an IF statement or logic gates- testing conditions).

Thinking ahead

Need to identify inputs and outputs required in solution before carrying it out. e.g. cooking- need to identify the different ingredients.

Gantt Charts:

Outlining tasks, how long they will take to carry out, and when they are carried out. Can identify and show what tasks can be carried out *concurrently*.



^ Bar chart for project schedule management (from markscheme). Time scale on top, list of tasks on side. Allows easy inspection and overlapping tasks, durations etc.

Pre-planning

Pre-fetching/ caching: Building libraries of pre-formed elements for future use, e.g. using Java libraries to increase efficiency, making sure you have your most commonly used spices ready at the front of your cupboard for cooking etc.
Pre-condition: Starting state before algorithm is executed, conditions that need to be fulfilled. e.g. have to have the required ingredients, a place to cook, pre-conditions while making decisions in

cooking (“Are the carrots still hard? Cook them a bit longer”).
Post-condition: Final state after execution of algorithm, the state you are trying to achieve/ lead up to, the final result.

Will need to also consider exceptions when building pre-conditions, e.g. identifying conditions for calculating end of year bonus when not all employees have worked for the company for the whole year.

Thinking concurrently

Concurrent processing: Implementing parts of a solution at the same time e.g. assembly line mass production- people carrying out task on one product then moving on to the next one while the next person carries out another task at the same time.
In computers:
Execution of different instructions simultaneously by multiple processors. Each processor processes different parts of a program's procedures and sub-procedures.

Needs better planning that accounts for different people working on the solution at the same time due to the changes they make, e.g. database should only be accessed once edit has been made otherwise the person wouldn't know someone else has erased their changes

Thinking abstractly

Selecting the pieces of information that are relevant to solving the problem and leaving out other information, to enable the ability to examine a solution at a human level of interaction. Considering something as its relevant characteristics and qualities, separated from concrete realities, actual objects or instances.
e.g.

- Map of London only showing roads and names, not the buildings because purpose is navigation along roads for cars
- Tube map showing simplified route as user is only interested in the order of stops
- Virtual reality games having smaller time scale and providing icons of items in inventories
- City simulation for pilots not having details like people or windows on buildings, just landmarks and the shape and height of buildings
- School decomposed into faculties
- An object in OOP is an example of abstraction because it hides the details of the code while preserving functionality

Program design

Array algorithms:

Sequential/ linear search: Usual search, go through every value and compare to the target value. Simple to implement, data doesn't need to be in order. Inefficient with large number of elements, may have to go through every single one of them.

Binary search: Values in order. Compare search value with middle value. If smaller, compare to middle value of sub-array to the left. If larger, compare to sub-array to the right, and so on. Faster than sequential search. Too complicated for small number of elements. Only works on sorted lists, difficult if data is constantly being added.

```
MARKS = [5,8,23,77,89,104]
TARGET = 89
MIN = 0
MAX = 5
FOUND = false

loop while FOUND=false AND MIN<=MAX
 MID = ((MIN+MAX)div 2)

 if MARKS[MID]=TARGET then
 FOUND = true
 POSITION = MID
 else if TARGET>MARKS[MID] then
 MIN = MID+1
 else
 MAX = MID – 1
 end if
end while
```

Bubble sort: Compare adjacent values. If not in order, swap round
Simple to write, less code. Takes more time to sort, average time increases almost exponentially as number of elements increase.

```
MARKS = [67,33,2,89,10,99]
TEMP = 0

loop X from 0 to 4 //4 = no. of elements-2
 loop Y from 0 to 4 – X
 if MARKS[Y] > MARKS[Y+1] then
 TEMP = MARKS[Y]
 MARKS[Y] = MARKS[Y+1]
 MARKS[Y+1] = TEMP
 end if
 end loop
end loop
```

Selection sort: Splits array into sub-arrays. First sub-array is sorted, second is unsorted. e.g. to sort in ascending order, find the smallest value, place it in the correct position in the first sub-array by swapping it with the element that was there, move position of beginning of sub-array forward one, loop through the rest (second sub-array) to find the smallest value again. Repeated for all elements.
Good for small lists. Not efficient with big number of items, have to find the smallest value many times.

```
MARKS = [67,33,2,89,10,99]
MIN = 0 //position of start of un-sorted sub-array
SMALLEST = 0 //position of currently smallest value found
```

```
loop MIN from 0 to 4 //loop to no. of values-2
 loop X from MIN+1 to 5 //loop through sub-array, MIN moves up 1
 if MARKS[X]<MARKS[SMALLEST] then //finds smallest value
 SMALLEST = X
 end if //by end of loop, position of smallest value found
 end loop
 TEMP = MARKS[SMALLEST] //swaps smallest value with value at start of sub-array
MARKS[SMALLEST] = MARKS[MIN]
 MARKS[MIN] = TEMP
end loop
```

Collection: Group of objects. No assumptions are made about the order of the collection (if any) or whether it can contain duplicate elements. We add and retrieve data from them.

Loops

Just know how to code the different types of loop tbh

Suitability of an algorithm

Efficiency: Amount of computer resources required to perform algorithm's functions ***Correctness:*** Extent to which algorithm satisfies specification
Reliability: Capability to maintain performance
Flexibility: Effort required to modify algorithm for other purposes

Big O notation: Measure of efficiency of an algorithm.
O(1) – efficiency and speed are always the same, time proportional to 1. e.g. addFront, algorithm that adds up fixed no. of values etc.
O(n) – time and efficiency proportional to *n*. e.g. linear search method (proportional to length of array, longer array = longer time searching, loop to non-constant value etc.) $O(n^2)$ – proportional to n^2. Time required increases rapidly if *n* increases e.g. nested loops in bubble sort and select sort

Nature of programming

Fundamental operations of a computer:

- Adding (and subtracting) values
- Comparing values/data
- Retrieving data
- Storing data

Compound operations use combinations of fundamental operations of a computer, e.g. “find the largest”

Conventions for pseudo code

- Variables in caps
- Keywords in lowercase e.g. loop, if
- Method names mixed e.g. getNumber
- Dot notation e.g. X.setNumber(1)

- // for comments
- Boolean operators in caps e.g. AND, OR
- loop X from 1 to 2, loop while, end loop, output, input
- =, >, <, >=, <=, ≠, mod (returns remainder e.g. 15 mod 7 = 1), div (how many times number fits e.g. 15 div 7 = 2)

Programming languages

Machine language: Low-level language directly understood by computer, made up of binary numbers.

Assembly language: Low-level language using symbols for instructions and memory addresses

High-level programming language: Uses elements of natural language. Easy to use for humans and more understandable. *Abstracts* some areas of computing systems, would otherwise take too long to write our systems in machine code.

Programming languages need to have...

- Fixed vocabulary- Instructions for operations do not change. e.g. “print” will always print
- Unambiguous meaning- Clear instructions
- Consistent grammar and syntax- The way we declare and use language features must be the same.
- Provide a way to define basic data types and operations on those types- ability to write functions/procedures
- It has to run on/be able to be processed by a computer- it must have a compiler/interpreter

Higher level programming languages can differ by…

- **Method of translation-** Whether by compiler or interpreter (or both)
- **Different programming paradigms-** e.g. procedural or object-oriented **Purpose of the language-** Specific for certain tasks (e.g. CSS for HTML websites or language for AI) or general purpose (can build any program with any logic e.g. Python) **Compatibility with different environments-** e.g. Java with virtual machine can run on all OS while some languages can’t
- **Syntax differences-** e.g. structure of statements

Source code: Original code/program developed using high level language. Needs to be translated into machine code to be run/executed by the computer.

Object/ target program: Program translated into machine language. Translation methods:

- **Compiler:** Executes translation process only once, translates the whole program. Object program is saved so it doesn't need to be compiled again. All errors are displayed when the whole program is checked, compilation ends only once errors are fixed. *Example:* C++
- **Interpreter:** Reads, translates and executes program line by line. Errors are displayed after each line is interpreted. Goes through the process every time the program is run, much slower than a compiler. *Example:* BASIC

Writing code

Variable: Used to store a data element that can be changed during program execution. Has an identifier and type.

Constant: Elements and quantities that don't change. e.g. final double PI = 3.14 ***Object:*** Comprised of data and methods (operations that can be performed by the object)

Use of programming languages

Advantages of breaking down into sub-programs

- Breaking complex job into simpler tasks
- Distributing program amongst different programmers
- Code reuse across multiple programs
- Reducing code duplication in program

Advantages of collections

- Methods are predefined algorithms, can immediately use
- Software reuse