**Cydney – Revision Notes**

**EXAM ONE – COMPUTER SYSTEMS**

**Systems architecture**

* Purpose of the CPU
  + Fetches instructions, executes them, returns the result
  + Two main components – CU (Control Unit), ALU (Arithmetic and Logic Unit)
    - CU is for timing, control, flow of data
    - ALU does arithmetic, logical processing
    - Can be divided into more detailed components
      * Registers – areas where values can be stored
        + E.g. MAR (Memory Address Register), MDR (Memory Data Register)
      * Program Counter (PC) register holds details of next instruction to be executed
      * Accumulator – register that holds values from the ALU
      * Cache – larger area where often- or recently-used values can be stored
  + Von Neumann architecture describes a computer and has 4 major components
    - CPU, memory, inputs, outputs
    - CPU works on fetch, decode, execute cycle
      * Fetch – get instructions from memory
      * Decode – turn compressed, encoded instructions into machine instructions
      * Execute – perform the instructions (e.g. add, jump, increment, move, call)
  + CPU performance affected by:
    - Clock speed (measured in GHz – how many instructions per second)
      * The smaller the clock speed, the fewer instructions can be executed, the slower the CPU can run, affecting program performance
    - Cache size (how large, in MB, the on-chip cache is, minimising calls to memory)
      * The smaller the cache, the more ‘cache misses’, the more calls to memory, the more the CPU waits for instructions and the slower the programs run
    - Number of CPU cores (affecting how many parallel operations can be performed)
      * A ‘core’ is simply another CPU – some CPUs have > 1 core, meaning there are many CPUs on the same chip enabling operations to happen in parallel
      * The fewer the cores, the less parallelism occurs, the more the CPU waits and the slower programs run
* Embedded systems
  + Two types of design:
    - RISC (Reduced Instruction Set Computer) – hardwired e.g. embedded devices
    - CISC (Complex Instruction Set Computer) – dynamic, like today’s PCs / mobiles
  + Why?
    - Some systems only need to do one job e.g. Internet-enabled washing machines, fridges, or drones
    - Some applications need custom computers e.g. aeroplanes
  + Fine line between embedded (RISC) and non-embedded (CISC):
    - Your phone is CISC – it’s a whole operating system of its own in a handheld device
    - Even though it isn’t a traditional PC, it is programmable and dynamic
    - An Internet-enabled doorbell, however, is RISC – one purpose and non-dynamic

**Memory**

* RAM and ROM – Random Access Memory, Read Only Memory
  + RAM is dynamic e.g. wiped and rewritten many times – used in CISC, used in PCs
  + ROM is read-only – programmed once, readable many times but not reprogrammable
    - Used mostly in embedded systems – RISC architectures
* Unit of measurement
  + Measured in bytes and their common multiples:
    - Bit – a single 1 or 0
    - Byte – 8 bits
    - Kilobyte (KB) – 1024 bytes
    - Megabyte – 1024 KB
    - Gigabyte – 1024 MB
    - Terabyte – 1024 GB
  + There is another system of measurement in use and commonly confused with the above
    - Kibibyte, mebibyte, gibibyte, tebibyte
    - Formally denotes exactly the same as the 1024 system above
      * Meaning byte, kilobyte, megabyte e.g. officially are multiples of 1000, not 1024
        + E.g. kilobyte – 1000 bytes
        + Megabyte – 1000 KB and so on
    - However this is a huge grey area…
      * The exam will be clear – if kibi, mebi etc. are specified you MUST calculate in 1024
      * Else calculate in 1000
      * You will likely get marks for either system but show your workings!
  + Memory runs at clock speeds too
    - Commonly measured in low GHz (or high MHz, same difference)
      * E.g. 2133MHz (or 2.3GHz)
    - Attempts to access the memory at a higher clock speed will cause damage
  + Some memory contains extra features
    - E.g. registers on the memory modules, not just on the CPU
    - E.g. local cache on the memory modules for faster access
* Virtual memory
  + Area on disk that acts like memory
    - Why? Sometimes main memory isn’t enough to hold all the required info
    - In this event the operating system can ‘spill’ data to disk
    - If there is a dedicated disk or area of disk for virtual memory, this can improve speed
    - Positive: More space for in-memory calculations
    - Negative: Can be slower as max speed limited by I/O interface
* Flash memory
  + Typically found in portable applications but not always
  + Common example – USB flash drives / thumb drives
  + Solid-state storage
  + Positives – Convenient, very fast
  + Negatives – Not very durable, portability comes at expense of security, can be expensive

**Storage**

* Purpose
  + To store large amounts of data that won’t fit in main memory
  + To store data that must be persisted when the computer is powered off
  + (For portable drives) To be able to move the data between different computers
* Types
  + Hard disk drives – rotational platters of magnetic disks arranged in spindles
    - The head moves towards or away from the centre as the disks rotate and reads the magnetic surface
  + Solid-state drives – non-moving solid memory chips arranged on a circuit board
    - Can be read and written to much faster than HDDs
      * Why? No moving parts ergo no rotational latency
    - But – more expensive than HDDs
    - Durability – they wear out much quicker
    - Evidence that they can be slower for sequential reads/writes than HDDs
    - Very expensive when compared to HDDs
  + Other types of storage
    - Tape storage (rare, now)
    - Online storage / cloud storage (storage via an Internet connection to another server)
    - Flash/portable/USB/thumb (many terms) storage – storage on a removable device
  + You will need to be aware of the following and how they apply to each storage type
    - Capacity
      * HDDs: Large, SSDs: Medium, Flash: Small, Tape: Very large
    - Speed
      * HDDs: Medium, SSDs: Very fast, Flash: Fast, Tape: Very slow
    - Portability
      * HDDs: Poor, SSDs: Poor, Flash: Very good, Tape: Good
    - Durability
      * HDDs: Good, SSDs: Poor, Flash: Good, Tape: Good
    - Reliability:
      * HDDs: Good, SSDs: Poor, Flash: Good, Tape: Good
    - Cost:
      * HDDs: Cheap, SSDs: Expensive, Flash: Very cheap, Tape: Cheap

**Wired and wireless networks**

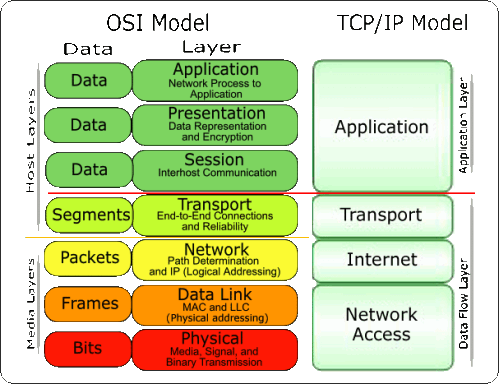
* Types of network
  + LAN (Local Area Network) – network connecting computers in the same geographical or logical region e.g. computers connected to the same network in your school computer lab
  + WAN (Wide Area Network) – network connecting computers in a wide region and often across network or geographical boundaries, may encompass several networks
* Performance
  + How do we measure performance?
    - Latency – how long it takes to get a signal from A to B (measured in milliseconds, ms)
    - Quality – how many data packets get through intact (as a percentage?)
  + What affects performance?
    - Distance from source to destination
      * Proportional to the speed of light, 186,000 miles per second
      * A second is 1000ms, so sending a data packet from the UK to Australia (9,500 miles) takes 51 milliseconds – an exchange of packets will take longer.
    - Interference
      * Can be affected by devices like microwaves, or solar radiation, or electromagnetic signals from other computer equipment
    - Congestion
      * If routers are busy, they may drop packets rather than queue them
    - Corruption
      * Miscommunication of headers or other packets
* Client-server network
  + Various computers that connect to servers and upload/download information from them
  + Central point of communication
  + Central, single version of the truth
  + Arranged in a star or snowflake-type formation
* Peer-to-peer network
  + Each computer is connected to one or more other computers in the network
  + No central point of control
  + Very resilient
  + But – difficult to keep files to the same versions and requires more maintenance
  + Less common than client-server
* Hardware
  + Wireless access points (a.k.a. APs) – devices through which you can connect to a network
    - E.g. your home router for your broadband
  + Router –
    - Routes data to and from connected devices and to/from the network it is connected to
    - Can modify, filter, retransmit that data – acts as a gateway
  + Switch –
    - Simpler device which routes data from one network to the appropriate connected computer and vice versa
    - Often a router and switch are built into the same device but…
      * Not always. Switches can be used especially when there are many devices
        + Switches can assign internal IP addresses
  + Modem –
    - Not the same as a router. Modulator/Demodulator (hence, ‘modem’)
    - Translates inbound and outbound digital signals between a copper phone line and the router
    - Very much a lower-level device responsible for translation, not logic
  + Network Interface Card (NIC)
    - This is the hardware device that translates data coming over a cable to something tangible
    - Every PC has one – NICs can be for wired, for wireless or both
  + Transmission media
    - Cabling, for example
    - But also WiFi channels
      * These channels are specific UHF (Ultra-high Frequency) radio channels
* Internet
  + Distributed collection of networks
    - Comprised of billions of devices
  + Addressable using Domain Name Servers (DNS)
    - DNS translates a domain name (e.g. [www.google.com](http://www.google.com)) to an IP address
    - Reverse DNS does the opposite, an IP address to a domain name
    - There are DNS servers which control ‘root-level domains’ – the last part of the domain (.com, .co.uk) and these are arranged into 13 groups
    - There are several hundred root-level DNS servers that serve these groups
    - Any organisation can host a DNS server at higher levels (the ‘google’ in the URL above)
      * Only registrars (registered with ICANN, a governing body) can approve ‘name’ servers which are DNS servers which reconcile your domain with your server IPs
  + Hosting
    - Servers which allow you to store files – either for use, or files which serve e.g. a website
    - Various sorts –
      * Your own hosting – you own the servers, you control them
      * Cloud hosting – you rent the hosting from another company/organisation and access remotely
      * Managed hosting – you pay another company to provide private hosting for you and manage the infrastructure
    - Cloud –
      * Large platform-as-a-service offerings allowing companies to quickly e.g. create services and provision storage without owning physical computers
      * These services are then available over network connections
      * Example: Backup services, so you can backup your computer to an online network location in case of disaster
  + Virtual networks
    - ‘Network-within-a-network’
    - Some terminology – VLANs (Virtual LANs), VIPs (Virtual IPs)
    - These are logical networks that don’t physically exist
      * Useful in some circumstances e.g.
        + For an office that wants a WiFi network for employees and another for guests

Both virtual networks share the same router and cabling and WiFi channels

However they are both arranged into two sets of IP addresses and computers on either VLAN cannot see or communicate with one another

**Network topologies, protocols and layers**

* Glossary:
  + Node – Any single element in a network that transmits or receives information
  + Client – A node in a network that receives information from another node
  + Server – A node in a network that transmits (serves) information to another node
  + Peer – Synonymous with node and used only in describing peer-to-peer networks
  + Topology – The design of a network (star, bus, mesh, grid, hybrid and so on)
  + Terminator – the end-point of a network (only applies to some network topologies like a bus)
  + ISP – Internet Service Provider. A company or service that provides connectivity to the Internet
  + Packet – A collection of data containing a header and a payload (user data) sent as a single unit on a network
  + Router – Acts as a gateway to an ISP, and routes signals from one input (the ISP) to multiple outputs (nodes)
  + Switch – Splits signals from one or more inputs to one or more outputs. Similar, but not the same as a router
  + Hub – Connects multiple points in a network and ensures all inbound packets are visible to all connected nodes
* Client-server and peer-to-peer – general ideas covered in the section above
  + Client-server advantages:
    - Centralised control of security policies, access and audit
    - Easy to find resources like files or database records
    - Good vertical scaling (basically, means we can upgrade the central server)
    - Easy to connect to other networks
  + Client-server disadvantages:
    - Susceptible to load problems (leading to denial of service – DoS)
    - Single point of failure
    - Central servers are expensive
    - Requires ongoing maintenance
  + Peer-to-peer advantages:
* Very resilient to failure – removal of any node does not damage the network
  + - No central server to maintain – reduces costs
    - Can sustain long periods of high load with redundant routing
  + Peer-to-peer disadvantages:
    - Difficult to manage files centrally
    - Lack of accountability (some peer-to-peer software is used for pirating)
    - Difficult to maintain software versions across all nodes (inconsistency)
* Topologies
  + Star and mesh – easy to remember shapes
    - Star – e.g. client-server
    - Mesh – e.g. peer-to-peer
  + Other sorts:
    - Snowflakes – star networks joined together
    - Bus – computers along a central backbone
* Ethernet
  + Primarily, this means cabling
    - CAT-5 – fibre-optic cabling connecting two computers or devices
    - Uses light transmission to signal 1s and 0s
    - Extremely fast
  + Interface is an ‘ethernet port’
    - Basic network port on the NIC
* Addressing
  + IP addressing – identifiers in the range 0-255.0-255.0-255.0-255 (this is IP version 4, or IPv4)
    - IPv6 has much longer hexadecimal-like IDs e.g. fe80::ccfb:d389:6ea0:1fe%21
  + IP addresses are unique network locations
  + MAC addressing – hardware IDs
    - Each MAC address denotes some piece of hardware
    - Not usually changeable
  + Transmission protocols (remember these):
    - What is a protocol? Like a language of communication between computers.
    - TCP/IP (Transmission Control Protocol/Internet Protocol) –
      * Handshake method of data transmission, used in Internet communications
    - UDP (User Datagram Protocol) –
      * Fire-and-forget type of data transmission, doesn’t require handshake, less common but still used especially in internal networks and with embedded devices
    - HTTP (Hypertext Transmission Protocol) –
      * The logical protocol used to fetch and render Internet assets, like web pages
    - FTP (File Transfer Protocol) –
      * The protocol used to transmit and receive files across network connections
    - POP (Post Office Protocol) –
      * An older form of transmission of e-mail messages (especially receiving)
    - IMAP (Internet Message Access Protocol)
      * The current protocol for transmission of email messages (especially receiving)
    - SMTP (Simple Mail Transfer Protocol)
      * The current protocol for transmission of email messages (especially sending)
  + Layers
    - OSI 7-layer model
    - You don’t need to remember the details but have a general idea of what components you can expect to find in each layer:



* Packet switching
  + Grouping data into packets
  + Each packet has a header and a payload
  + The packets are sent as datagrams
  + The datagrams are read as a stream of data
  + The method of reading the data depends on the transmission protocol
  + Data is routed across a network depending on the information in the header
    - E.g. destination IP address

**System security**

* Why?
  + To protect the confidentiality (how secret is it), integrity (how intact and reliable is it) and availability (how available for use is it) of the systems and the data
* Glossary:
  + Attack surface – All aspects of a particular IT system that are subject to threats
  + Threat – a specific technique or method of mounting an attack
  + Attack – An undesirable event initiated by a malicious actor with the intent to breach the confidentiality of the system, disrupt the integrity of a system or affect the availability of a system.
  + Malicious actor – a person or organisation that carries out, or attempts to carry out, an attack
  + Confidentiality – The ability to keep secret information about an organisation’s IT assets, or the data contained within their IT assets
  + Integrity – The ability to have trust in the accuracy, validity and reliability of an organisation’s IT assets
  + Availability – The ability for IT systems to be available for use
  + Risk – Normally defined as a function of likelihood and impact with respect to the probability of an attack and the damage such an attack could cause.
  + Likelihood – The probability that an attack could occur
  + Impact – A measure of disruption to the organisation should an attack occur. This can be measured in financial terms, or as a measure of time, or other ways such as whether the organisation can continue.
  + Vulnerability – Something about the IT system that enables an attack. Not the same as a threat – a threat is something that could happen, a more general class of attack. A vulnerability is the thing that enables a threat, or allows an attack to happen.
* Forms of attack
  + Confidentiality –
    - Hacking to obtain data, breaking-and-entering, theft of data or system access without the owner’s consent
    - Can be done locally (on-site, access to computer) or remotely (breach of security defences)
  + Integrity
    - Intentional corruption of data
      * E.g. malicious employee looking to cause business disruption
    - SQL injections to cause damage via web forms
  + Availability
    - DoS (Denial of Service) and DDoS (Distributed denial of service) attacks
      * Flooding a website / server with requests so it cannot cope with legitimate traffic
      * Particularly via botnets e.g. computers infected unknowingly with malware
* Threats
  + Malware – software with a malicious intent
  + Phishing – Trying to defraud people through pretence or deception e.g. with a fake website
  + Social engineering – Convincing people to give up passwords etc. or gain access via con-man skills
  + Brute force attacks – e.g. finding a password by trying all combinations
  + Denial of service – as above, flooding a server or service with requests
  + Data interception and theft – e.g. through rogue (pretend) access points
  + SQL injection – forcing a web form or other interface to return information it was not designed for
  + Poor network policy – ability to use network design to compromise systems e.g. find unprotected shared drives on a network
* Vulnerabilities
  + Not the threat but the thing that enables the threat to become a reality
  + Weak spots in systems which enable a threat to be carried out
  + We try and identify these and take measures to prevent threats from happening:
    - Penetration testing – internal tests by companies/organisations to identify vulnerabilities
    - Network forensics – taking logs, monitoring network traffic, identifying malicious actors
    - Network policies – policies that apply to many computers, set centrally e.g. IP blacklists
    - Anti-malware software – software (a.k.a. anti-virus) that is installed to detect and block installation of malware whether intentional or not
    - Firewalls – devices (virtual or physical) that enable or disable communications from different IP address ranges between one network and another
    - User access levels – a.k.a. the principle of least privilege – users should only have the permissions to perform those actions that they are meant to perform, and no more
    - Passwords – use of strong passwords that aren’t susceptible to brute-force attacks
    - Encryption – encoding data so that it cannot easily be decrypted by somebody without the proper key

**System software**

* Operating systems (OSs):
  + Embedded
  + Windows
  + \*nix
  + Mac OS
  + Other (rare)
* Glossary:

|  |  |
| --- | --- |
| User interface | The means by which a user interacts with the operating system. This can be graphical (GUI), via the command line (Command Line Interface, or CLI), or even through more esoteric means – voice, for Alexa and related devices, and touchscreens. |
| Operating system | The collection of software that abstracts the hardware in a computer system to a usable, universal software interface to enable a user to gain value from it. |
| Kernel | The interior of an operating system – the core. Normally incorporates the underlying programs, drivers, HAL and/or controllers. |
| Driver/Controller | Software responsible for communicating with a hardware component or peripheral. For example, a driver than can communicate with a printer. |
| Library | Collection of methods that can be used by a program during run-time. On Windows systems these are often labelled .dll. |
| Peripheral | Some hardware component that is plugged into the main computer system, such as a printer, speakers or a monitor. Requires compatible drivers. |
| Shell | A CLI (command-line interface) session that allows the user to use the operating system of the computer it is connected to. Not necessarily the computer from which the user is working. |

* User management – creating and managing users of an operating system
* File management – systems for managing files accessible by an OS
  + E.g. hierarchical, via a folder or directory
  + E.g. more complex linked systems, like Sharepoint
  + Symbolic links (a.k.a. shortcuts) enable file access from different locations
* System software (utilities):
  + Defragmentation – physically rearranging files on disk for more sequential reads and therefore faster access (less rotational latency)
    - Less-used for SSD and other non-HDD storage as provides little advantage
  + Encryption software –
    - For encrypting data. Common standards:
      * MD5
      * SHA-1
      * AES
    - These are one-way algorithms. Using a public key a message is encrypted
      * Only the private key can encrypt it
      * This security is ensured by prime number factorisation
        + (In short – it is very easy to multiply two large prime numbers and obtain a third number, but very hard to take the third number and work out which two primes were used to calculate it)
        + This process is the Diffie-Hellmann algorithm
      * Also known as public-key cryptography
    - Where the keys are identical, known to both parties this is symmetric key algorithm
  + Compression
    - Achieved using mathematics e.g. compressing a string of 0s to a single 0 and the number of times it occurs
    - Or achieved using lossy techniques –
      * Approximating the ‘average’ colour of a group of pixels to save on the amount of data that needs to be stored
    - Lossless data compression is more difficult as information must be retained
      * Focus on efficiency
  + Backup
    - Full
      * Full backups of a data set or database – from top to tail
    - Differential, or incremental
      * Backups of only the data that has changed since the last full backup
    - Full backups are good for guaranteeing the data is preserved in full
      * But takes a long time to restore
    - Incrementals are very fast to take
      * But require more than 1 plus the last full backup to restore
      * Take less space than full backups

**Ethical, legal, cultural and environmental concerns**

* You may face some questions in the exam about cultural and environmental concerns:
  + These are blind e.g. we don’t know exactly what will be asked
  + Bear in mind the current zeitgeist of environmental / climate change concern…
    - Computers use rare earth minerals and metals which take energy to mine
    - There are human rights concerns over these mining activities
    - There is a carbon footprint associated with manufacturing and transporting computer equipment
    - Computers use a lot of energy. This energy is derived mostly from fossil fuels.
    - Computers radiate gases like ozone into the atmosphere.
    - Recycling – very difficult in practice and some components are poisonous
  + Cultural implications…
    - Positive implications – education for all
    - Freedom of communication, connection, ideas sharing
    - However – proliferation of e.g. terrorism, easier to organise malicious groups
    - Spread of material for minors – appropriate concern over safeguarding
    - Spread of memes or tropes which are inaccurate e.g. anti-Semitism/Islam propaganda
    - Spread of political propaganda, disinformation
  + Use your common sense if asked a question in this area and try to reference current events
* Legal and ethical
  + Content and software licensing – e.g. Creative Commons and open-source/vs. proprietary
    - Creative Commons has several letter abbreviations
    - You don’t need to know these or their combinations but you ought to know…
      * That you can share your work and allow amendments (or not), attribution, and use in commercial activities.
    - Open source – developed for use by all, normally without charge, sometimes with open-source licenses
      * E.g. the GNU General Public License
    - Proprietary –
      * Owned by an organisation
      * Subject to the terms imposed in the license that the organisation issues
      * Usually (but not always) paid-for software
  + The Data Protection Act and GDPR
    - Data Protection Act 1998 and 2018 (which one, not sure which is tested)
      * 1998: Data must be:
        + Fairly and lawfully processed
        + Processed for limited purposes
        + Adequate, relevant and not excessive
        + Accurate
        + Not kept for longer than is necessary
        + Processed in line with your rights
        + Secure
        + Not transferred to other countries without adequate protection
      * 2018: Five principles:
        + Data processing should be fair, lawful and transparent
        + Personal data shall be collected for specified, explicit and legitimate purposes
        + Personal data must be adequate, relevant and limited to what is necessary
        + Personal data shall be accurate and kept up to date
        + Personal data shall be kept for no longer than is necessary
      * GDPR (DPA 2018) rights of the Data Subject:
        + The right of transparency and modalities
        + The right to be informed
        + The right of access
        + The right to rectification
        + The right to be forgotten
        + The right to restrict processing
        + The right for notification obligation
        + The right to data portability
        + The right to object
        + The right in relation to automated decision making and profiling
  + The Computer Misuse Act
    - Unauthorised access to computer material
    - Unauthorised access with intent to commit or facilitate the commission of further offences
    - Unauthorised modification of computer material
    - Plus amendments e.g. for terrorism
    - Also creation or possession of articles to facilitate an offence as above (Section 3A)
  + Copyright law
    - The Copyright, Designs and Patents Act (1988)
    - Moral rights:
      * The right to be named as the creator of the work (‘paternity’)
      * The right to object as someone named wrongly as the creator of the work (‘false attribution’)
    - Automatic (in the UK)
    - US equivalent – Digital Millennium Copyright Act (DMCA)
    - Normally 70 years protection with exceptions

**EXAM TWO – COMPUTATIONAL THINKING, ALGORITHMS AND PROGRAMMING**

This exam is about your ability to think and reason in the same terms as a computer does, using techniques rather than learned knowledge.

You will have practiced some of this with your programming project.

Below is the exact exam specification.

Use this exam specification to go through each point and draft out specific examples of each area you’re not sure about, looking up information from our past sessions and from the Internet where you’re not sure.

Next to some sections I’ve included URLs for you to investigate for specific examples or more explanation of what each term means.

Use the past papers provided – take this exam as if for real, then mark your answers according to the marking scheme. Use the examiner’s report to see common mistakes. You’ll be able to see the level of detail needed and the common kind of questions you might be asked.

**Algorithms**

• Computational thinking:

⃝ abstraction

⃝ decomposition

⃝ algorithmic thinking

• standard searching algorithms:

<https://www.geeksforgeeks.org/searching-algorithms/>

⃝ binary search

⃝ linear search

• standard sorting algorithms:

<https://www.geeksforgeeks.org/sorting-algorithms/>

⃝ bubble sort

⃝ merge sort

⃝ insertion sort

• how to produce algorithms using:

⃝ pseudocode

⃝ using flow diagrams

• interpret, correct or complete algorithms.

**Programming techniques**

• The use of variables, constants, operators, inputs, outputs and assignments

• The use of the three basic programming constructs used to control the flow of a program:

⃝ sequence

⃝ selection

⃝ iteration (count and condition controlled loops)

• The use of basic string manipulation

• The use of basic file handling operations:

⃝ open

⃝ read

⃝ write

⃝ close

• The use of records to store data

• The use of SQL to search for data

• The use of arrays (or equivalent) when solving problems, including both one and two dimensional arrays

• How to use sub programs (functions and procedures) to produce structured code

• The use of data types:

⃝ integer

⃝ real

⃝ Boolean

⃝ character and string

⃝ casting

• The common arithmetic operators

• The common Boolean operators

**Producing robust programs**

• Defensive design considerations:

<http://teach-ict.com/2016/GCSE_Computing/OCR_J276/2_3_producing_robust_programs/defensive_design/miniweb/index.php>

⃝ input sanitisation/validation

⃝ planning for contingencies

⃝ anticipating misuse

⃝ authentication

• Maintainability:

⃝ comments

⃝ indentation

• the purpose of testing

• types of testing:

⃝ iterative

⃝ final/terminal

• how to identify syntax and logic errors

• selecting and using suitable test data.

**Computational logic**

• Why data is represented in computer systems in binary form

• Simple logic diagrams using the operations AND, OR and NOT

<http://www.ee.surrey.ac.uk/Projects/CAL/digital-logic/gatesfunc/>

• Truth tables

• Combining Boolean operators using AND, OR and NOT to two levels

• Applying logical operators in appropriate truth tables to solve problems

• Applying computing-related mathematics:

⃝ +

⃝ –

⃝ /

⃝ \*

⃝ Exponentiation (^)

⃝ MOD

⃝ DIV

• Characteristics and purpose of different levels of programming language, including low level languages

• The purpose of translators

• The characteristics of an assembler, a compiler and an interpreter

<https://www.geeksforgeeks.org/language-processors-assembler-compiler-and-interpreter/>

• Common tools and facilities available in an integrated development environment (IDE):

⃝ editors

⃝ error diagnostics

⃝ run-time environment

⃝ translators

**Data representation**

• Units - bit, nibble, byte, kilobyte, megabyte, gigabyte, terabyte, petabyte

• How data needs to be converted into a binary format to be processed by a computer.

• How to convert positive denary whole numbers (0–255) into 8 bit binary numbers and vice versa

<https://www.electronics-tutorials.ws/binary/bin_2.html>

• How to add two 8 bit binary integers and explain overflow errors which may occur

• Binary shifts

• How to convert positive denary whole numbers (0–255) into 2 digit hexadecimal numbers and vice versa

<https://www.wikihow.com/Convert-from-Decimal-to-Hexadecimal>

• How to convert from binary to hexadecimal equivalents and vice versa

• Check digits.

• The use of binary codes to represent characters

• The term ‘character-set’

• The relationship between the number of bits per character in a character set and the number of characters which can be represented (for example ASCII, extended ASCII and Unicode)

• How an image is represented as a series of pixels represented in binary

• Metadata included in the file

• The effect of colour depth and resolution on the size of an image file.

• How sound can be sampled and stored in digital form

• How sampling intervals and other factors affect the size of a sound file and the quality of its playback:

⃝ sample size

⃝ bit rate

⃝ sampling frequency

• Need for compression

• Types of compression:

⃝ lossy

⃝ lossless