1. Digital Devices

1.1 Classifications

Understand, be able to define, and be able to give examples of these categories of contemporary digital devices:

- ***Computers***
  A computer is an electronic device that can perform arithmetic and logical operations, store and retrieve data, and execute instructions. It consists of hardware components such as a central processing unit (CPU), memory, input/output devices, and storage devices. Computers can be classified into various types based on their size, functionality, and purpose.
- ***Mobile Phones***
  A mobile phone, also known as a cell phone, is a portable electronic device that allows users to make and receive calls, send and receive text messages, and access the internet. It consists of hardware components such as a display screen, battery, processor, memory, camera, and input/output devices. Mobile phones can be classified into various types based on their features, operating system, and form factor.
- ***Embedded Systems***
  An embedded system is a computer system that is designed to perform a specific task or function within a larger system. It consists of hardware components such as a microcontroller, sensors, actuators, and input/output devices. Embedded systems can be found in various applications such as automobiles, medical devices, home appliances, and industrial equipment.
- ***Storage Devices***
  A storage device is a hardware component that is used to store and retrieve digital data. It can be either internal or external to a computer system. Examples of storage devices include hard disk drives, solid-state drives, USB flash drives, and memory cards.
- ***Peripheral Devices***
  A peripheral device is a hardware component that is connected to a computer system to enhance its functionality. Examples of peripheral devices include printers, scanners, keyboards, mice, and webcams.

This is but *one* way to classify devices, and that there may be other categories, such as *wearable devices*, *gaming consoles*, *smart home devices* like those that "powered by" Alexa (Amazon Echo), Siri (Apple devices), and Microsoft's Cortana.

1.2 Functions and Features

Understand the difference between a feature and a function:

- ***function***
  Something useful that a device can be used for. Example functions of a smart phone include video calls, browsing the internet, and game play.
- ***features***
  The tools that accomplish functions. The features of the smart phone that enable to function of video calls include the front-facing camera, microphone, and speaker.

Take some time to consider different *features* of the digital devices that you use, and how these *features* enable the *functions* that you use each device for. *Features* to consider include:

- *portability*
- *performance*
- *storage*
- *user interface (UI)*
- *connectivity*
- *media support*
- *energy consumption*
- *expansion capability*
- *security features*

1.3 Technologies

1.3.1 Processors

CISC: x86 (AMD, Intel)
RISC: ARM, RISC-V, Apple's M2, M3,...

1.3.2 Memory / Storage

Volatile memory is referred to as ***RAM***. When we say volatile, we mean that the data is lost when power is removed from the system. There are a few types of ***RAM***:

- ***SRAM*** (*Static Random Access Memory*)
  Uses a six-transistor memory cell similar to a flip-flop; frequently used for CPU cache.
- ***DRAM*** (*Dynamic Random Access Memory*)
  Uses a transistor-capacitor pair as a memory cell. The memory cell requires refreshing as the capacitor loses charge over time. Most user-replaceable memory modules use *DRAM*.

There are also various types of ***non-volatile memory***, including:

- ***ROM*** (*Read-Only Memory*)
  Stores data permanently and cannot be modified after it is created. It is used to store firmware and other programs that are essential to the operation of a computer or other electronic device.

- ***PROM*** (*Programmable Read-Only Memory*)
  Can be programmed once by the user. Once programmed, the data stored in PROM cannot be changed or erased.
- ***EPROM*** (*Erasable Programmable Read-Only Memory*)
  Can be erased using ultraviolet light, and reprogrammed multiple times
- ***EEPROM*** (*Electronically Erasable Programmable Read Only Memory*)
  Can be erased and reprogrammed multiple times using an electric charge; this means it can be re-written by circuitry within the electronic devices.
- ***Flash Memory***
  Is simply a type of EEPROM. The name is meant to make it seem fast.
- *SEEP*
  Serially accessed EEPROM often used to store small amounts of data.

1.3.3 Connectivity

USB (Type A, B, mini, micro, C; 2, 3,0, 3.1, 3,2, 4), Thunderbolt, Firewire, Ethernet, Wifi, Bluetooth, NFC, Cellular, ...

1.3.4 Global Positioning System (GPS)

1.3.5 Quick Response (QR) Code

1.3.6 Radio Frequency Identification (RFID)

1.3.7 Near-Field Communication (NFC)

1.3.8 Other Technologies

Know the basics of these technologies:

- *biometrics*
- *battery power*
- *touch screen*
- *miniaturization*
- *sensors*

1.4 Technological Convergence / Digital Convergence

The A-Level curriculum uses the term “technical convergence”, but this doesn’t seem like the proper term to me. At least remember the word ***convergence***. If you study biology, this process is similar to *convergent evolution*.

These terms refer to the integration of different technologies into a single device or platform. Examples include smartphones, smartwatches, all-in-one printer/scanner/copiers, etc.

1.5 Performance

Understand factors that can be used to assess performance of digital devices:

- *speed*
- *capacity*
- *portability*
- *bandwidth*
- *power efficiency*

2. Software

2.1 Types of Software

There are two main categories of software: ***systems software***, which includes the *operating system*, and ***application software***.

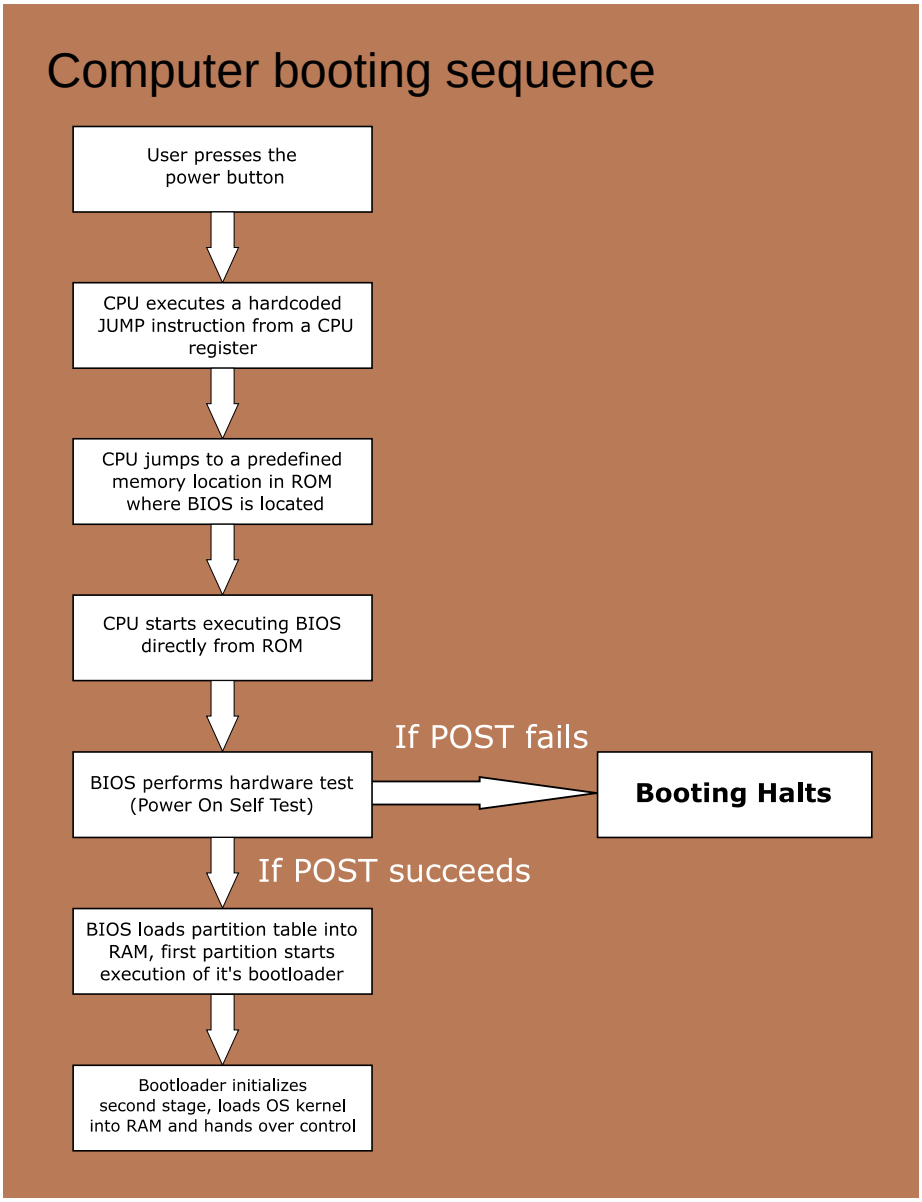
2.1.1 Systems Software

Systems software is software responsible for managing and controlling the hardware components of a computer system. It provides a platform for application software to run on. Examples of systems software include:

- ***Firmware***
  Firmware is software that is embedded in hardware devices. It provides low-level control over the device's hardware components. Examples of firmware include BIOS or UEFI, and firmware for routers, modems, and storage devices.
- ***Boot Loader***
  The boot loader for a computer is stored in the boot sector of the storage device. It is started via *firmware*, and are responsible for loading and initializing the operating system kernel (core program) during startup. Examples of boot loaders include *GRUB* (GRand Unified Bootloader), *LILO* (LInux LOader), and *NTLDR*, and the *Windows Boot Manager.*
- ***Operating System* (*OS*)**
  The OS manages computer hardware resources and provides common services for computer programs. Examples include Microsoft *Windows*, *macOS*, and *Linux* on PCs and *Android* and *iOS* on mobile phones.
- ***Device drivers***
  A device driver allows the operating system to communicate with a hardware device. Examples include printer drivers, graphics card drivers, and sound card drivers.

- ***Utility software***
  Utility software is designed to help manage, maintain, and optimize the computer system. Examples include antivirus software, disk defragmenters, and backup software
- ***Programming Language Translators***
  Software that translates written code in one programming language to another programming language. Examples include compilers, interpreters, and assemblers.

The diagram below shows steps for a computer to boot. The processor control passes from *firmware*, which is written for the specific hardware it is running on, to the *boot loader*, and finally to the *operating system*.



Consider the role of the operating system in managing:

- devices
- processes
- users
- security

2.1.2 Application Software

Application software is designed to perform specific tasks or functions for the user. It runs on top of the *operating system* and relies on the services provided by the systems software. There are numerous ways to categorize application software into sometimes overlapping categories such as *productivity software*, *office software* (Microsoft Word), *communications software* (Skype), *educational software*, *entertainment software* (Steam, Netflix), *media software* (Adobe illustrator, VLC Media Player, iTunes), d*atabase software* (MySQL), etc.

2.2 Software Licenses

2.2.1 Distribution and Use

These categorize how a license restricts use and distribution

- ***Proprietary***
  Used for software that is owned by a company or individual. It typically restricts the use, modification, and distribution of the software.
- ***Free***
  Allows users to access, modify, and distribute the software without any restrictions. The users may or may not have access to the source code.
- ***Open-source***
  Allows users to access, modify, and distribute the source code of the software. There are many different types of open-source licenses, each with its own specific terms and conditions. ***Permissive*** licenses (MIT, Apache) allow users to modify and distribute the software without restrictions, as long as they give credit to the original author. ***Copyleft*** licenses requires that any software derived from the original code must also be made available under the same license, ensuring that the software remains open-source and freely available (GNU Public License).
- ***Creative Commons***
  Creative Commons is a U.S. non-profit corporation that released a number of licenses to facilitate the free distribution of an otherwise copyrighted work. Where open-source allows for collaboration and development of software, Creative Commons licenses allows creators to specify the conditions of use of other artistic works such as music, images, and videos. Restrictions, for example, may include non-commercial use and attribution to the creator.

2.2.2 Licensee

These categorize the which users are permitted to use the software.

- ***Single-user***
  Allows one user to install and use the software on a single device.
- ***Multiple-user***
  A license for more than one individual. This includes licenses for a specific group or team, as well as broader ***institutional licenses*** for an entire company or university.

2.2.3 License Model

- ***Named User***
  Licensed to a specific individual who may be able to install or use on multiple devices. This could be implemented with an activation code or cloud-based.
- ***Per Device / Per Seat***
  Licensed to a specific piece of hardware. Any number of users may use the software, but only on the specific device to which the license is assigned.
- ***Concurrent / Floating***
  A set number of copies of the software can be in use at once. This is implemented using a license server. This is usually implemented using a licensing server; Pearson may refer to this as a ***network license***.
- ***Site / Enterprise***
  An unlimited number of copies may be used by anyone within a defined physical location or entire organization.

2.2.4 Duration

Software licenses may be categorized based on their duration

- ***fixed-term***
  Allows use of the software for a certain period of time. If it is expected that user will renew the license at the end of the term, it may be referred to as a ***subscription*** license, and technical support and upgrades are usually included. Another fixed term license is an *evaluation* or *trial* license, which is typically very short-term, generally on the order of a few weeks.
- ***perpetual***
  Allows use of the software indefinitely. Technical support is usually limited and generally upgrades are not included.

2.3 Software Maintenance

Upgrade versus patch; Compatibility (backwards compatibility); Automatic upgrade

3. Data Representation

3.1 Number Systems

3.1.1 Binary

A base-2 number system that uses only two digits, 0 and 1, to represent numbers. Each digit is called a "***bit***," which is short for "binary digit." The word binary is derived from the Latin prefix "bi-" which means two. The binary number system is essential to computer science because digital devices frequently store and process data using electronic circuits that can be in one of two states, on or off, which correspond to the binary digits 1 and 0, respectively.

3.1.2 Denary (Decimal)

The decimal number system (often referred to in A-Level papers as the denary system) is a base-10 number system that uses 10 digits, 0 to 9, to represent numbers. It is used in everyday life and is the most commonly used number system. In computer science, decimal numbers are often converted to binary or hexadecimal numbers for processing.

3.1.3 Hexadecimal

A base-16 number system that uses 16 digits, 0 to 9 and A to F, to represent numbers. It is used in computer science to represent binary numbers in a more compact and readable format. Each hexadecimal digit represents four binary digits, which makes it easier to work with binary numbers. Hexadecimal numbers are often used in computer memory addresses, color codes, and other applications.

3.1.4 Octal

A base-8 number system that uses 8 digits, 0 to 7, to represent numbers. It is used in computer science to represent binary numbers in a more compact format. Each octal digit represents three binary digits. Octal numbers are not as commonly used as decimal or hexadecimal numbers, but they are still used in some applications such as file permissions in Unix-like operating systems.

3.2 Data Sizes and Rates

3.2.1 File Size

3.2.2 Data Transmission Rate