AS ICT

Module Two

Revision

Just the facts

Systems Software

controls the functioning and management of a computer's resources

A ***device driver*** is a **program** that **controls** input / output / storage devices ***(peripherals).*** They allow ***configuration*** of the peripheral and ***communication*** of ***instructions and messages*** in the **correct *(translated)* format** between the OS and the peripheral.

They can be **supplied on a disk** with the product or **built into the OS**, supporting ***plug and play (when device drivers are automatically loaded when a new peripheral is installed)***

System Utilities are programs that **perform** a **specific** common **function**:

- **User Management** ***(to audit users' activities and allocated IDs and passwords)***
- **Virus Scanners** ***(which check memory for viruses & malware and delete them)***
- **File Management** ***(which can repair, compress, backup and defragment files)***

The ***Operating System*** **controls** and **monitors** all **applications**, and **provides** an **interface** between the **user, software and hardware**. Four popular ones are:

- **MS Windows** ***(provides a GUI and was originally based on DOS)***
- **MS-DOS** ***(is command-driven)***
- **Linux** ***(open-source rival to windows)***
- **Mac OS X** ***(works only on Apple hardware - used in design industry)***

The Functions of the OS

Allocates **memory** to running **apps**	**Logs** and presents **errors**
Schedules / prioritizes **processing** tasks	**Regulating access** and users
Manages input / output **devices**	**Interrupt** processing

Modern operating systems can *multitask*

This gives the **illusion** of **lots** of **programs running** at the same time when the **CPU** is just **dividing** its **time** between them.

Operating systems can be:

- **Single-user** (where only **one** user has access at a time)
- **Multi-User** (where **one computer** can be accessed by a **number of users** at once)
- **Distributed** (where a **number of computers** are used to run an application)

The Human Computer Interface

*The **User** and **Computer** can **interact** through the **interface**:*

- **Users** can **instruct** the **computer** ***(through an input device)***
- The **computer** can **respond** and **prompt** for input ***(through an output device)***

There are **three** main types of **HCIs:**

Command-Line	***Menu-Driven***	***Graphical User Interface***
A **simple prompt** into which the user types **commands**. It can be extremely **powerful** and **quick** if you **memorise** all of the **commands** but **no good** for **average** computer users	This guides the user through a **set** of **menus** to provide **access** to **functions**. This is **more accessible** than command line but functions can be **buried** in **layers** of menus.	These consist of: • **Windows** - all input happens in self-contained "windows" which can be manipulated by itself • **Icons** - these provide intuitive graphical representations of applications of commands • **Menus** • **Pointers** - a mouse (or other) allows the user to control the others **GUIs are intuitive** *(you can **guess** what to do in an new situation)* and there is often **contextual help** but requires a lot of **processing power** and many **commands are hidden** from novice users.

Most applications now have a ***common UI***. They have the **same keyboard shortcuts** and **menu structure**, so they are **easier to learn.**

Natural Language interfaces **understand human speech**. It ***should be*** extremely **easy to use**, and **good** for **disabled people**, but it's **processor-intensive**, people have **different accents** and use **different phrases** and some people **don't like computerised speech**.

Hallmarks of a good UI:

Easy to use ***(tailored to user's level of expertise)***	Keeps the user **informed**	Is **consistent** and **intuitive**
Customisable	Be **predictable**	**Respond** in a reasonable time
Be clear and **logical**	Require **minimal effort**	**Validate** user input
Give lots of **help**	Commands easy to **remember**	Takes **Health & Safety** into account

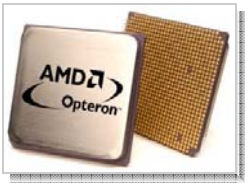
- **Colour schemes** and **layouts** that are **easy on the eye**
- **Less typing** ***(pre-emptive text entry)*** and **clicking**

Software Upgrade and Reliability

You may want to ***upgrade*** your system for a number of **reasons**:

- Some components are **not functioning adequately**
- There is a **need** to upgrade to **support other** components
- New, **beneficial products** come onto the market
- There is a **financial** or **business incentive** to buy

Speeding up processing



- **Increase clock speed** ***(of processor)*** but this will only make the processor work **as fast** as the **slowest component** in the system
- Add a **co-processor** so that **specialised tasks** can be executed **faster**
- Add a **parallel processor** so that **overall processing** is **quicker**

But eventually, **systems** will be **replaced** with **newer, better systems**.
This will have some ***issues***:

New hardware	***New software***
• It might not work with **existing software** • Users will need to **learn** to use it • Some **familiar functions** may be **unavailable**	• It might not work with **existing hardware & other software** • Users will need to **learn** how to use it • **Unfamiliar** and **missing** features will lead to **mistakes** and **wasting** of **time** • New systems may cause **stress** and **dismissal** of **staff** that were needed to maintain old system

But **new systems** must be ***reliable***, so we must **test** it to **ensure** that it **functions correctly**, **performs within acceptable limits** ***(time and volume of data processed)*** and **recovers from system failure**.

Exhaustive testing covers **every possibility**, and to ensure this we construct a **test plan** which contains a **schedule** and **details** of the **tests** with **what** they'll test, the **data** used and the **expected results**. ***This is like in your coursework.***

After this testing, there is **alpha testing** ***(where restricted internal use is carried out with feedback)***, **beta testing** ***(where a few members of the public use it extensively and feedback)*** and finally **acceptance testing** ***(to prove its suitability)***

Common Applications

Generic Software is for everyday use and can do lots of tasks (i.e. spreadsheet packages)

Task-specific software is usually for a specialist application (i.e. CAD software)

Organisations will usually use a **combination** of **task-specific** and **generic** software *(perhaps in an **integrated package** or **suite** like **Microsoft Office** so they have **common features**)*

Also, they may have software especially written for them: ***bespoke software*** ***(as opposed to off-the shelf software)***

Bespoke software...

Fits **purpose precisely** and **performs exactly** as **needed**	Extremely **expensive** compared to off-the shelf
Likely to use **available resources** more **efficiently**	May take a **very long time** to **produce**
	Testing is **limited** and **support** and **upgrades** are **not** widely **available**
	All **staff** will need **training**

Good software has a **range of features**, is **upgradeable**, **compatible** with other software ***(data portability)*** and **hardware**, and **customisable**

Types of **generic software**:

- **Database** Packages ***(to systematically store, search and sort data in related tables)***
- **Spreadsheet** Packages ***(to display and process data for analysis and modelling)***
- **Word Processors** ***(to create text-based documents)***
- **Presentation** Software ***(allow users to give multimedia presentations)***
- **Desktop Publishing** ***(to create documents with complex layouts integrating graphics and text)***

Object Linking and Embedding *(OLE)* allows **information** in **other formats** to be ***shared*** between applications and ***integrated*** into a document.

- A ***linked object*** is when there is just a **link** to an **existing file** in your document. This will **update automatically** with **changes** to the **original** file.
- An ***embedded object*** is when there is a **copy** of the **object actually saved** in your document. **No changes** to the **original** file are **reflected** in the embedded object.

Other features include ***macros*** which allow you to **automate repetitive tasks** and ***application generators*** which **generate code** with a **minimum** of **technical knowledge** through the use of **wizards** ***(i.e. database programs with wizards)***

Generic Applications can be ***customised*** to fit users' requirements using:

- **Buttons** *(to execute a **command** or for **navigation**)*
- **Forms** *(to **input data** using features like **drop-down menus** and **validation** built-in)*
- **Menus** *(to give the user a **logical choice** of **actions** and make **interaction and navigation** easier, either **pop-up, pull-down or full-screen**)*
- **Macros** *(to **automate tasks** by **recording** a **series** of **commands**)*

It's **faster** and **easier** to **enter data**, means **less mistakes**	**Options** may be **left off** and buried in menus = **less powerful** interface
Data can be **validated** on **entry**	**Normal** users cannot **sufficiently customise** the interface to benefit
Tech support is **easier** as the **complex** interface is **hidden**	The interface may not work on **upgraded** software versions

Templates and Wizards

Templates are **preset documents** which contain **basic structure** and **formatting** which the user can use as the **base** for their own **customised documents.** Users can even **create** their own, **custom templates**.

(i.e. pre-laid out letters with logos and formatting already applied)

Users don't have to **waste time formatting** documents	If **generic** templates are **no good**, users **waste time** creating their own
Consistent and **professional** house style	Documents can become **boring** and **repetitive**

Desktop Publishing uses ***style sheets (master documents)*** which store far **more information** such as the **position** and **formatting** of **individual** page **elements**. Usually, someone is **paid** to develop the style as it is quite a **creative task**.

Wizards also help you carry out **certain tasks easily** and **without much effort**. You choose from **preset layouts** and **styles** to create a **complex documents** or **applications** without **technical knowledge** but you're **limited** by the **options available** and the **end result** often looks **similar** to other wizard-produced documents.

Processing Data

Reservation and **booking systems** use **Transaction Processing**.

Transaction Processing is when **each individual transaction** is **processed before** any **subsequent transactions**. All transactions must be **checked** and **correct** before being **submitted** as there is ***no feedback*** until the entire transaction has been completed.

This means that in a seat booking context, no seats can be double-booked since one reservation (transaction) must be dealt with before the next.

Transaction-based systems may then have an ***interactive front-end*** which **responds** to **actions** from the user and provides the missing **feedback**. Information can be **retrieved** and checked before the **final transaction** takes place.

Data can then be processed by...

Real-Time Processing, where **stored data** files are **updated regularly** enough so that there is **no apparent delay** for the user. All **information** is kept **up-to-date at all times** and the system ***responds immediately to inputs***. However, it is often **expensive** to set up.

Batch Processing, where transactions are **collected over time**, and then the **main data** is ***updated in one go***. This **reduces load** on **resources** by running at weekends, but the system is **not flexible** and the data is **not kept up-to-date** at all times.

All data is stored as 1s and 0s called **binary digits *(bits)***, but as long as the **application reading** the data knows what to **expect**, it will be **interpreted in the correct format**.

Computers can process various types of data including:

- **Text** - usually stored using **ASCII code** where a **set of bits (7 or 8)** represents a **character**
- **Numbers** - stored using **binary representation** for **integers**, **floating point representation** for **real numbers**, and **two's complement representation** for **negative numbers**
- **Graphics** - **raster images** are stored as a series of **pixels**, each represented as a **set of bits**. **Vector graphics** are stored as **data** about the **lines** and **shapes** that make up the image.
- **Sound** - sets of bits represent a sound's **amplitude** and **frequency**.

128	64	32	16	8	4	2	1
0	0	0	1	0	1	1	0

An example of **binary representation *(00010110 = 22)***

Data needs to be **encoded** using a **Standard Format** so that it can be used by **many applications & systems**. There may be a number of **different formats**, so the computer is **informed** of a file's format by its ***extension*** **(i.e. .doc for a Word Document)**

Formatting Data

A system is of **no use** to the **end-user** if it does not produce ***useful information***.
To be useful, information must:

- Be in a **suitable format**
- Be easily **accessible** and **not buried** in lots of other information
- Be **presented appropriately**
- Tell the user exactly **what they want to know**

When thinking about the **format** of the output information, consider:

- **Who** needs the information ***(and in what format)***?
- **What** bits of information do they need?
- **When** do they need the information ***(regular updates or when complete)***?
- Will the information be needed for **another part of the system**?
  Will it need to be stored in a particular format, or sent over a network?

Output can be **presented** as:

- **Tables** - to put information in **order** and make it **easy to read**
- **Graphs** - to display **statistical data** in an easy to understand **visual** context
- **Graphics** - a picture speaks **a thousand words**, easier to **react** to quickly
- **Reports** - a **combination** of the above that **only contains required info** and uses features such as **grouping** and **clear labelling** to enhance its **usefulness**
- **Multimedia** - is more **captivating** and **entertaining**

The **format** that information is presented in depends on its ***target audience***. A **sales director** may want the **figures** in a **table** for **analysis** but the **shareholders** would want a **summary report** or **slideshow** with **graphics**.

Producing the output to the **specification** as **easily** as possible can be achieved by using:

- **WYSIWYG editing** *(where you can **see exactly** how your document will look in its **final format** while you are **creating** and editing it, or using a **print preview**)*
- **Mail Merge** *(where one document can be used as a **template** for many **personalised** ones with **details** stored in a **database**, so each letter is of an **identical standard**)*

Also, **don't forget** that very often output is simply ***saved to disk*** *or* ***sent to another system*** via a network. In this case, it is **important** to use **standard file formats** to ensure that the file can be **opened** by **other users** and **applications**. Output can also be ***displayed on a screen***, but for **large screens** or **projections**, the output must be of a **high resolution.**

Input Devices

All systems need ***instructions***, which generally come from the **user** via an ***input device***

*Input devices are **chosen** for their **quality** and **ease of use***

Keyboards are used to enter **alphanumeric data** and some **commands.** ***Some systems have specialised keyboards i.e. mobile phones.***	People are **familiar** with their use and they are **very versatile**, and often have **function keys** to quicken commands	**Transcription errors** are very common and they can cause **RSI**, and the user's typing **skill** can influence **input quality**. They're also quite **big**.
Mouse / Trackball / Trackpad (laptops) is used to **move** the on-screen **cursor** in a GUI and transmit **click** commands	People are **familiar** with them and they are easy for **beginners**. Also, many **GUIs require** their use.	Excessive use can cause **RSI** and giving **commands** can be **slower** than using a keyboard. They also may degrade over time as **dirt** is **picked up**
Scanners are used to **capture** hardcopy **images** and **text** into a **digital format**	They are the best way to get **documents** onto a computer **close to** their **original format, faster** than **typing out** documents	**Special software** is required to perform **OCR**, which is often **inaccurate**
Touch Screens are used to interact with a **GUI** ***(replacing the mouse)***	Suited for use in **public places**, as **hardware** can't be **stolen** and a **surface** for **mouse** use isn't needed. Also **user-friendly**.	It is **difficult** to **pinpoint** screen items **accurately**, and leads to **slow text entry** *(as an on-screen keyboard must be used)* and **limited input** capabilities.
Concept Keyboards use **sensitive surfaces instead** of **keys** to input certain **commands** or items of **data**	Very **easy** for **young** and **inexperienced** users and may **speed up** data entry as it's **customised**. Can also be **cleaned easily** (of dirt & liquid)	It **cannot** be used for **many purposes**, as it has been **specialised** to one application
Graphics Tablets are used to input line drawings using a stylus. **Light Pens** draw directly onto a screen.	**Drawings** can be **traced over**, so **accurate reproductions** are produced. Similar to **real-life drawing**, so it's **intuitive**.	Depends on the **skill** and **steady hand** of the user, and **larger** sizes can be quite **expensive**.
Digital Cameras can **capture** an **image directly** into **digital format**	They are **familiar** to users and produce **digital formats directly**	But the **quality** of the image may **not** be **as good** as with a **traditional camera** and the **files** may be **large**
Card Readers ***(Magnetic / Smart)*** extract data from **strips** or **chips** on **plastic cards**.	It is much **quicker** and **accurate** than **inputting** personal details **by hand**	But they can be **forged** and **damaged easily** and **magnetic data** can become **corrupted** over time
MICR *(Magnetic Ink Character Recognition)* is used in **banking systems**	Speeds up cheque **processing** and makes **forgery harder. Humans** can read it too (unlike barcodes)	They are **not used** anywhere else due to having **limited applications**
OMR *(Optical Mark Reader)* reads **marks** on **specially designed forms** (lotto tickets)	Is a **quick** and **accurate** way of entering a **large amount** of **pre-formatted** data	The **range** of **responses** are **limited** and **forms** must be **designed** and **filled in correctly**
Bar Code Readers use **lasers** to read a **printed bar code**	Provides **extremely fast** and accurate **input** of data	But can **only** input **bar codes** and nothing else.

There are also some **special** input devices for **disabled people**:

- **Foot-mice**, identical to a **mouse** but **designed** to be **controlled** using the **feet** for those with **RSI** or other **debilitating illnesses**
- **Voice recognition** allows **commands** and **dictation** to be given to software but at the moment it is **inaccurate** due to **ambiguities** in language and **accents**. They also require **training**.
- Various devices for those with **little motor control** *(cursor position and on-screen keyboard controlled by **eye movement**)*

In some **control systems** designed **without human intervention**, inputs can come from ***sensors*** *(such as **heat/light/sound** sensors)*. The system will then **react accordingly** to these **environmental inputs**.

Output Devices

Printers are one of the most **popular** types of **output devices**:

- **Inkjets** **spray** ink onto **paper** to **create** the **printout**. They're **cheap** and produce **adequate quality** documents, but they're quite **slow** and the **ink** can sometimes be **expensive**.
- **Laser Printers** use a **laser** to etch **electrostatic charge** on a **rotating drum** to **transfer ink powder** to **paper** and then **fuse** it. They're **very fast** and **reliable**, and produce **very high quality output**. But they are much more **expensive** than inkjets and **toner** cartridges are **costly**

Plotters are used in **design offices**. An **arm** moves over the **paper dispersing ink** or **tracing** with a **pen**. They produce **very accurate graphics** and handle **large paper sizes**.	**Dot Matrix Printers** are used **sparingly** nowadays ***(for receipts mostly)*** and are **impact printers**, that print a **series** of **dots** that look like **letters** at a **distance**. They can be used to make **carbon copies**.	**Thermal Printers** are often used in **fax machines**. They **"burn"** an image onto **special** thermal **paper**.	**Braille Printers** create **raised dots** on paper for **visually impaired** users

You can decide on a printer by comparing their **speed**, **output quality**, **bulk printing capacity** and **initial** and **running costs**.

Monitors ***(VDUs)*** present **text** and **graphics** using **small dots** of **light** ***(pixels)***. They are often the **most used** output device. They differ in:

- **Colour capabilities** (monochrome or full colour)
- **Size** (measured diagonally in inches)
- **Maximum resolution** (in dots per inch or pixels)

CRT monitors use up far **more space** and **electricity** but are **cheaper** and offer **better colour reproduction**

LCD monitors are **light** and use up **less space** and **electricity**, but have a **limited viewing angle** and **"refresh rate"**, and are **more expensive**.

There are also **projectors**, which act as **monitors** but **project** the **screen** onto **another surface** ***(usually used for presentations)***, and **speakers**, which provide **audio output** from the system and are an **integral** part of a **multimedia** system.

Backing Storage & Backups

Computer systems need to **store data** so that it can be **reused**. This is done by keeping it on ***backing storage***. There are 3 kinds of storage device:

- **Magnetic** storage - where **magnetised areas** on a **magnetic surface correspond** to **individual bits**. A **read/write head** is **used**.
- **Optical** storage - where **high-power lasers** are used to **burn "pits"** into the **surface** of the medium, and also to **read** them **back**.
- **Solid state** storage - this memory uses **semi-conductors** to hold data **permanently**. They are **non-volatile**.

The **main storage media** are:

Hard disk, which is the **main** storage medium for **PCs**, can store a **massive amount** of **data** and allows **fast random access**. They can **crash** after **long periods**.	**Floppy Disks** have **low capacity** and are **quite slow** so are used primarily to transfer **small files** between systems as they are **highly portable** and **cheap**.
CDs have a **high storage capacity** and allow **faster** random access than **floppies**. They are becoming **cheap** and home **CD-Writers** allow users to take advantage of them.	**DVDs** have **very high storage capacity** and **access speeds** and offer all of the **benefits of CDs**. **DVD writers** are also becoming available.
Magnetic Tape has a **capacity comparable** to **hard drives** and are usually used for **backup** as they **do not support random access**.	**Flash Memory** is **becoming cheaper** and is in some cases **faster** than a **hard drive**. **Large capacity** means it is suited for **backups** and **file transfer**.

Normally the **hard disk** is used as the **primary backing store** while **backups** are made onto **secondary storage**. **Programs** can usually be **reinstalled** at any time but **data** is sometimes **irretrievable**, so **regular backups** should be made. There are a number of backup methods, including:

- **Mirrored Hard Disks** ***(RAID systems)*** where a **second** hard disk mirrors the first. This is extremely **reliable** and means the **disk** can be **used immediately** but it's very **expensive**.
- **Global Backups** where you copy **all of your data in one go**. This is a **necessary** step in most backup systems and **reinstating** a system from it is **quick**, but **backup itself** is quite **time consuming**, especially when repeated.
- **Incremental Backups** where you **only copy** the **files** that have **changed** since the last backup. This **saves** a lot of **time during backup** but **reinstating** the system **takes** a **long time** as **files** have to be **rebuilt**.

Security of Data

Data **Privacy** is the **need** for **some *(personal)* data** to only be **held** and **accessed** by **authorised** people

Data **Integrity** is ensuring the **correctness** of data at **all stages** of **processing**

Ensuring **privacy** and **integrity** as well as **preventing loss** of data is **data security**

Computer systems and the **data** that they contain are **very valuable** to companies, and so need to be ***protected***. **Personal data** is protected by the **DPA** and **losing** it or **exposing** it to **alterations** could land the company in **legal troubles**.

Accidental loss of data can occur due to **human error**, **natural disasters** and **problems** with the **hardware** and **software**.	**Deliberate** loss of data occurs due to **cyber-crime**, **sabotage**, **viruses** and **vandalism**.

Hardware can be kept secure by:

- Restricting **physical access**
- Installing **Alarm** Systems
- Installing backup **batteries**

Software can be kept secure by:

- Setting certain files/folders **read-only**
- **Write-protecting** disks
- **Encrypting** sensitive data for transmission
- Using a **firewall** and **virus** scanning software
- **Password**-protecting systems
- Giving users different **levels of access**
- Using an **audit trail** to pinpoint attacks

Password Tips:

0. Use a **mixture** of **numbers** and **letters**
0. **Don't** use a **real word**
0. Keep a **secure record** of passwords
0. **Change** them **regularly**
0. Use **different passwords** for things

However, even with **all these steps** it is **inevitable** that some data will be **lost** or **corrupted**. To **combat** this, we can ***backup*** data so that data can be **restored** if needed.

Backups should be:

- **Frequent** enough so **data** is **up-to-date**
- Able to be **quickly restored**
- Kept **secure**, in a **remote**, safe **location**
- **Checked periodically *(to ensure that the procedure works)***

*Some companies also have **replacement hardware** on **another site**, so that their **business** can **keep running** in the **worst case scenario***

Networks

Computers can either work **stand-alone**, or in a **network** with others. The **definition** of a network is *a **number** of **computers linked together** allowing **resources** to be **shared** and **data** to be **transferred** between them.*

There are **good** and **bad** things about networks:

• **Hardware** can be **shared** • **Software** and **data** can easily be **shared** and **transferred** • A user can **access files** from **anywhere *(hot-desking)*** • **Data management** (i.e. backup) can be **centralised** • **Access** to **certain facilities** (i.e. the Net) can be **controlled**	• You need **extra equipment** and **staff** to build and oversee networks • If there is a **fault** in the network, you may **not** be able to **access** your **files** • Users have **less control** over their system setups • **Networks** can be **slow** if many people are using them at the same time • **Viruses** can **spread** easier via networks

And there are **two types** of network:

LANs (i.e. school network)	**WANs** (i.e. the Internet)
Spread over one **small geographical area**	Can be spread over a **large distance** (ie towns or countries)
Connected using **cables** or short-range **wireless**	Connected using **telephone** or **satellite** connections

Networks can be **linked together**:

- A **bridge** is used to link **two similar LANs**
- A **gateway** joins a **LAN to a WAN**
- **Routers** determine the **best route** for data
- A **repeater** amplifies signals on **long-distance LANs**

And **individual** computers can be **joined** using:

- **Copper** cables (i.e. CAT5)
- **Fibre-optic** cables
- Microwave / **satellite links**
- **Telephone** lines

A network requires some **hardware/software**:

- A **network interface card** for each computer
- A central **server** with a **networking OS** and **management software *(optional)***
- A **modem / gateway** to connect to a **WAN**
- Browser and e-mail **software**

And the network must follow **communication protocols** such as TCP/IP to ensure compatibility

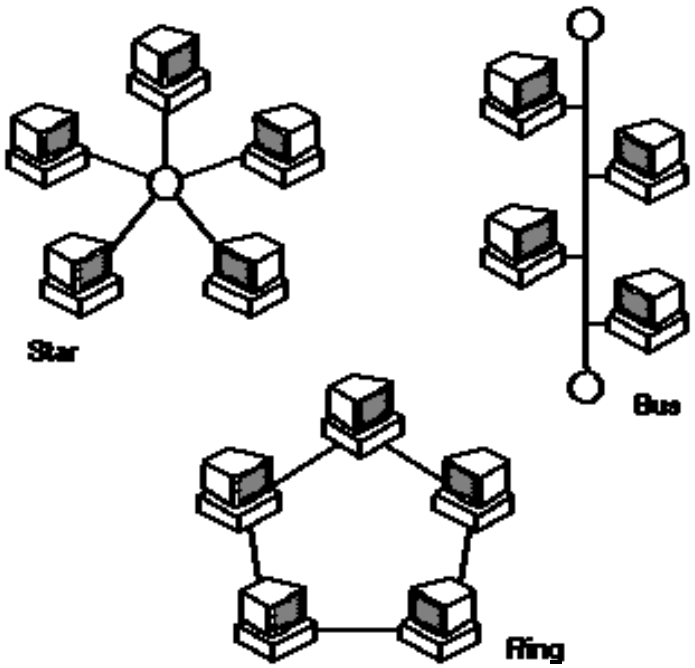
We measure networks' **bandwidth**. This is the **maximum** amount of **data** that can travel over it in a **given time period (not the speed of the connection). Broadband** allows **many signals** to be transmitted **at once**, and so has a **large bandwidth**.

Mobile computing is now possible where individuals connect to **networks** using a **wireless card**. They can be used **anywhere** within range of an **access point** and **no cables** are needed but transmission speed is **not as fast** as a wired network and **laptops** are usually **more expensive** than desktops.

Network Topologies

There are **two types** of LAN:

- **Peer-to-peer** - when a **small number** of computers **communicate among** each other **without** using a **server**. These are **easy** to **set up** and **cheap**.
- **Client-server** - when a **main server** controls the network and may **store** all **files** / **resources**. It requires **specialist software** and is more **expensive**, but **powerful**.



Bus Networks

- **One long cable** with computers, servers and peripherals attached along it
- Each **connection point** is a **"node"**
- Only **one signal** can travel **at a time**, so **quite slow** under stress
- **Cheap** because **little cabling** and **easy** to set up
- If **one break**, the whole network will **stop working**.

Star Networks

- All computers connected to a **central hub**
- Each computer has its **own connection** to the server
- It's **more expensive** than a bus network, more cabling
- Very **dependent** on the **central hub** or **server**
- It's **faster** than a bus network
- The network is **more fault-tolerant**
- It's **easy** to add **extra computers**

Ring Networks

- Data only **flows one way** around the ring
- Always a **peer-to-peer** network (no server)
- It is **cheap** and **quite fast** (unless there is **lots of traffic**)
- **Not dependent** upon one computer
- If **one node** malfunctions, **whole system** can go down

Servers can control the following:

Storage of users' **files**	**Network** printing	**Access** using user IDs and passwords
Automatic **backup** of files	Storage of **shared apps**	**Auditing** of user actions

Data Capture

Data Capture is **inputting** data into the computer in the **best format**. There are:

Manual Data Capture:	**Automatic Data Capture:**
• Collected using **paper-based forms**	• Collected using **sensors** or **scanners**
• Entered into the computer by a **human**	• **No** need for **human** intervention
• Often **cost-efficient** and most **practical**	• **Reduces** transcription **errors**
• Prone to **human error** while keying in data	• **Speeds up** data entry
	• May **not** provide **perfect** entry (ie OCR)

Data capture devices are listed in the Input Devices section

But it's **very easy** to put **incorrect data** into computer **systems**. Data needs to be **checked** using **Validation (is it reasonable?)** and **Verification (is it correct?)**.

Invalid data is **impossible** or **unreasonable** (i.e. a birth year of 9999)
and **Inaccurate** data is **valid** but **not true or correct.**

When data is captured **manually**, **incorrect data** can enter the system at:

- **Data Capture** - forms are **wrongly filled in** or **illegible**
- **Transcription** - **typing errors** or entering a form twice
- **Transmission** - **corruption** while being transmitted
- **Processing** - **problems** with **hard/software** may introduce errors

There are a number of **validation checks**:

- **Presence** check
- **Range** check
- A **Check Digit** *(a **digit calculated** from **all** the **other digits** in the piece of data)*
- **Cross-field** checks *(where the **value** in **one field** is checked **against another**)*
- **Drop-down** lists
- **Control totals** ***(where a real total is checked i.e. money owed)*** and **hash totals** ***(where an nonsensical total is checked i.e. phone numbers)***

But it **relies** on the **integrity** of the **software's** error-trapping and **only** makes sure that the data is **reasonable**... **not correct**.

There are a number of **verification procedures:**

- **Proofreading** - **comparing entered** data with **original** data
- **Double entry** - **entering** the **same data** in **twice**, by **different people**. **Errors** are **highlighted** and **addressed**.

But these are **time-consuming** and **expensive**, and only makes sure that **data matches** its **source**... **not** that it's **correct**. **Human error** is also still possible.

Databases

A **database** is a **collection of related data** and can store **lots of it**. They can **search**, **sort** and **combine** the data and produce **reports**.

Anything we store data about is an **entity** and each entity has **attributes**. In a database, data on the **entity** is stored in a **table**, and the **attributes** becomes **fields**. Each **record** in a table stores data about **one particular item** (an **instance**) and must be **uniquely identifiable**. The **field** that makes the record **unique** is the **primary key** and the **value** of this for **each record** must be **different**.

The **simplest** databases consist of **one table** and are called **flat-files**. They are used to store **lists** and are very **easy** to create, but contain a lot of **duplicate data**. They **waste** a lot of storage **space** because of this, and **data** can be **inconsistent**.

Databases with **more** than **one table** are **relational databases**. Each item of data is stored **once**, there is **no redundancy**. This **speeds** up **data entry** and **reduces inconsistencies** and **space wasting**. Relational databases use a **DBMS** which **can control access** to the database and **enforces** referential integrity (**consistency**). But they are **complex** and **time-consuming** to set up and run. Tables are linked through **relationships**. The linked field is called the **foreign key**.

One-to-one (i.e. **one company car per employee**)	**One-to-Many** (i.e. **one** library **member** can take out **many books**)	**Many-to-many** (i.e. **many customers** can buy **many products**) - *these are **best avoided** by using a **link entity** (in this case, **orders**. **One customer** can have **many orders** and **one order** can have **many products**)*

There are also **Hierarchical Databases** where the **records** are **linked** together in a **tree structure**, but they're **not used** much any more because they are **restrictive**.

Searching databases for items that **meet certain criteria** is called **querying**. **SQL** is a **language** that lets you **choose data** to **extract** from **tables** to **construct** a **query** but it's a bit **complicated**, so there is a **system** called **QBE *(query by example)*** that lets you use a **simple grid** to **specify** search **criteria** and **fields** in a **visual** manner. **Queries** on the database can be **performed** by **any application** through the **DBMS**. There are three main query types:

Parameter queries *(select queries)* search a **particular field** for a **value** and return all **records** containing the value. They can either be **static *(have a fixed parameter)*** or **dynamic** *(where the **user inputs** the parameter)*	**Complex** queries search using **more than one parameter** using logical functions such as AND/OR/NOT	**Cross-tab** queries **collate** queried data together into **a grid**, with various **field names** as row and column **headings**.

But **queries** are **returned** looking just like **tables**, they're **not formatted**. **Reports** condense the data in queries (and tables) by **grouping** and **summarising** it into **readable**, **formatted documents**. They can also present data as **graphs**.

Standard Database Table Notation

PRODUCT_TABLE(ItemCode, Description, Price, *SupplierCode*)

The **primary key** is **underlined**, and the *foreign key* is written in *Italics* **(or double-underlined when handwritten)**. When **handwriting**, always give a **legend** to **describe** your **method** so the **examiner** can't take away marks for **non-standard notation**

Entering and Retrieving Data

Data entry screens should be **designed** to make it ***easy*** to enter data into a database, so **errors** are **reduced**. As a rule, **forms** with **instructions**, **buttons** and **sensible error messages** are **easier** to use than just entering data **straight** into a ***table***.

Consistency

- **Screens** should be **similarly designed**
- **Buttons** should be in the **same place**
- **Data** entered on a **number** of **screens** should have the **same format** each time
- **Screens** must **match the layout** of the **paper form**

Relevance

- The **system** should only ask for **relevant data**
- **Instructions** must be **concise** and **useful**
- **Animation** and **bright colours** must be **kept to a minimum**

Sources of information can be:

- **Static** - **unchanging** and most probably **accurate**, but may be **out-of-date** and **limited** in scope ***(i.e. in a CD-ROM encyclopaedia)***
- **Dynamic** - **updated regularly** and large **volumes** of info available, but **no checking** process may have taken place ***(i.e. the Net)***

Searching

CD-ROMs have a **searchable index**, while you can use a **search engine** to find information on the **internet**.

The **Internet** has an **advantage** over **CD-ROMs** as **lots** of **information** can be found from **differing sources**, and **any errors** found can be **addressed quickly**. But you can **never** be **sure** of the **accuracy** of the information you find on the Net...

Information on the **Internet** can be ***filtered***. **ISPs** can **remove illegal sites** from their servers, and **schools** may **block access** to sites if their **content** is **inappropriate**

Modelling Software and Spreadsheets

A ***computer model*** consists of a **set** of **data** on **a real-world phenomenon** and **rules** which **control** the **behaviour** of the **model**. It is a **simplified representation** of the object.

There are **two** basic types of modelling:

- **Modelling of Objects** ***(i.e. CAD)*** - this creates a **virtual representation** of a real-world object so that the **design** can be **tested**, **experimented** on or **changed** easily.
- **Financial Modelling** *(done with **spreadsheets**)* - easy to perform ***what-if analysis*** where certain **parameters** are **changed** and the **effects** can be seen in **real-time**

Spreadsheets are made up of:

- **Cells** - a **data store** that holds **one piece** of **data**. Cells can be **individually formatted** and **protected**, and **conditional formatting** applied to them
- A **range** is a **group** of **cells** which can either be **named** or just **identified** by **cell references** ***(i.e. C5:D8)***
- **Rows & Columns** - **rows** are identified by **numbers** and **columns** by **letters**, then can be **hidden** and their **size altered**
- **Worksheets** - a **single page** of a **spreadsheet** that can be **named** individually and given different **access rights**. Usually contains **data** on **one specific area** or topic.
- **Workbook** - a **collection** of **sheets** which makes up the **entire spreadsheet** document

Spreadsheets have:

- **Formulas** - these are **instructions** to the computer to **process** the **data** held in certain **specified cells** and automatically **update** when **data** is **changed**. ***(i.e. =C3+D5*76)***

  Formulas can have **absolute** ***(C4)*** or **relative** ***(C4)*** cell **references**, so that when they are **copied** into **other cells** ***(replicated)*** they will either **stay the same** or **change** to reflect the change in position respectively. You may want to use **absolute** cell references when there is a **constant value** that you want in **all formulas** regardless of position.

- **Functions** - these can be **part of formulas** and are **standard routines** built into the spreadsheet that perform a variety of **complex** and **useful functions** ***(i.e. AVERAGE)***
- **Variables** - these allow **cells** to be **identified** by **reference** or **name** for use in **formulas**. They **hold data** about the **model**.

Computer Models have many ***advantages***:

Computer models are **portable** and **interactive**, they can be **changed without** extra **capital investment**	They are **safer** than conducting **real tests** on things ***(i.e. blowing up a building, medical testing)***	They can **perform** to a **high degree** of **mathematical precision**, which would take **lots** of **time** for a **human**
The model is **dynamic**, and **reacts** to **changes immediately** and **graphically**	Computer-based models can be **timeshifted** so that **intricate details** can be **witnessed**	But **some aspects** of objects just **need real-world** testing, such as a **customer** sitting in **a chair** for **comfort**

Customising Spreadsheets

Spreadsheets show a **load** of **figures** which are not that easy to understand.
Charts show the figures in a **easy to understand** and **visual** way.
*There are **many different types** of chart, so it's **important** to choose the most **appropriate** graph for your data.*

- **Bar Charts** are **useful** when **comparing two sets of discrete data *(ie sandwich sales per day)***
- **Line graphs** show **change** in **continuous data** over a **given time period**
- **Scatter graphs** show the **correlation** between **two sets of data**, and a **trend line** can be plotted to **accentuate** this trend.
- A **pie chart** is **divided** into **segments** which show the **contribution** of an **individual value** to a **whole**.

Form controls can be used to **customise** a **worksheet** and make **data entry *easier.*** There are many, including:

- **Buttons** - the **button** can be **linked** to a **macro** so that a **function** is performed or the user is taken to **another worksheet**
- **Option Buttons** - the user can select **only one option** from many
- **Checkboxes** - the user can turn on or off **many options at once**
- **List Boxes** - this contains **a list of items** that the user can **select** from
- **Combo Boxes** - this is like a **drop-down list box** that the user can either **select one item** from or **type** their **choice** into
- **Labels** - these add **instruction text** to the worksheet to **help** the user

Macros let you **automate repetitive tasks** by **recording** a **series** of **commands** into **one** button-activated **unit**. ***They can print the worksheet, perform some tidying up on the sheet, or anything else!***

Word Processing and DTP

Both types of application support **templates**, allow you to **format** the **layout** of your **documents** and **insert graphics**, **tables**, **etc**.

Word Processors are more suited to **text-based documents**: they often have **detailed dictionaries** and **spelling/grammar checkers**, **automatic indexing** for research, and **ready-made templates** for standard documents.

They also support ***mail-merge*** to allow **one standard letter** to be produced, and then **personal** data **integrated** into it ***(from a database)*** to produce **many personalised letters**. **Merge fields** are **inserted** into the document **where fields** from the **database** will show up, and **extra commands (word fields)** can be used to **control** what data is merged and **prompt** the user for extra data. This **saves time** and means **extra proofreading** is **not needed**.

DTP software has lots of **features** that make **complex layouts achievable**:

- They are **frame-based**, meaning that **text** and **images** can be easily **shuffled around the page** and **resized** in their frames **without affecting** other page elements.
- **Frames** can also be placed in **different layers** so **some things** are **on top** of others.
- **Master pages** can be used so that a **consistent look** is achieved for the whole document
- They allow **positioning** on a **grid**, or with **guides**, so that frames are positioned **accurately**
- They also have a **workspace** around the page where **items** can be **temporarily stored** without printing or being part of the final document.

Standard documents are **common** documents like **letters**, **memos**, **faxes**, etc which **don't** need a **specific application** to produce. You can **change** the **orientation**, **margins** and **size** of a **document** and apply **formatting** to individual **characters** or entire **paragraphs**. **Breaks** can also be inserted to **start** a **new section**, **column** or **page**. **Graphics** can also be inserted and **manipulated**, in the form of **images**, **clipart** or **autoshapes**.

Headers and footers can also be added to the document so **some information** is **displayed** on **every page**, such as automatically updating **page numbers**. **Footnotes** *(at the bottom of a page)* and **endnotes** *(at the end of the document)* can also be added to **explain items** in your document.

All of these **features** are used **to one end**: *to make the document suitable for your audience!*

For example, **children** will need **big fonts** and **colourful layout** while **older readers** will want a **clear**, **uncluttered** document.

Graphics Tools

Clipart are **graphics** that have been **released** into the ***public domain,*** so that **anyone** can use them **without paying royalties** or a **license**. You either get them **free**, or on a **CD-ROM** with a **browser** for flicking through them. They are also available in **topical image libraries** on the same **theme**.

• Clipart is **free** or **very cheap** • It's **better** for **non-artists** who cannot produce their own graphics • **No special training** or **investment** is needed in **illustration programs** or **scanners** • A **wide variety** of images are available	• Clipart **does not cater** to **every interest**, it is **limited** • **Searching** may take a **long time** to find the required clipart • Clipart is **unoriginal** and some is of **bad quality**

But **clipart** images can also be **edited** to ***customise*** them or ***animate*** them

Images are stored in one of **two main formats**:

Bitmap images - stored as a series of **pixels** which together make up the entire image. **Editing** them involves **changing individual pixels**, but packages like **Photoshop** make it **easy** and **fast** and have lots of **effects** to enhance images. Bitmaps are **good** for **complex** and **photorealistic** images but the **file sizes** can be **huge** unless the files are **compressed**. When the **image** is **enlarged** they become **pixellated**.	***Vector*** images are saved as **geometric equations** that **represent objects** in the image. Some **office programs** have vector graphics capabilities **built-in**. The objects can be **manipulated individually** and the **files sizes** are much **smaller** than bitmap images. They can also be **resized infinitely** but they require a lot of **processing power** to **display** and **manipulate**.

Presentations and Web Pages

Presentations used to be given using **OHPs** where **transparent slides** were used. These had to be **changed by hand** and could **easily be damaged**, and the **speaker** would have to use **makeshift methods** to **cover up** any **information** he wanted to reveal. Also, **no sound** or **animation** could be used, meaning the presentation may have been quite **boring**.

Multimedia presentations use a combination of **graphics**, **text**, **animations** ***(of both text and graphics)***, **video clips** and **sound** to keep the audience ***interested***. Slides can have **eye-catching transitions** and **hyperlinks** and **image hotspots** can be inserted so that **users** can **navigate** around the presentation **themselves**. All of the slides in a presentation should follow a **similar design scheme**, and presentation software allows this by providing **pre-defined colour schemes and master layouts**. Some presentations can even be **self-running**, advancing by themselves **without** having to have a **speaker** to present them.

• Presentations look **professional** and catch people's **attentions**	• It's easy to get **carried away** with multimedia
• Presentations can be **saved** and **adapted** for different audiences easily	• The software needs **expensive hardware** (a projector) and a powerful computer

---

Websites are written in **HTML**, but you *don't have to learn it* to make a **webpage**.

- **Web-authoring software** usually has a **WYSIWYG** interface so you can **visually construct** your pages with the help of **wizards** and website **management tools**.
- You can also integrate **dynamic elements** like **Java** or **Flash** into your page to **spice it up**, but they require that users have the correct **plug-in**.
- However, **users** will have to **learn** the **functions** of the package and they may be quite **expensive**.

Using a **word processor** or **DTP package** to create a **website** **doesn't** require lots of **technical ability** but they end up looking quite **messy** because there are **problems** in the **conversion** to **HTML**. Also, they often produce **larger**, **inefficient pages**.

Web pages are then **viewed** using a **browser**, which renders the HTML code as **viewable pages**. **Hyperlinks** take viewers to **different pages**, and the **back** and **forward** buttons can be used to **navigate independently**. The users can **bookmark** your site for **later visits**, or type the **URL** into the address bar.

Answering Exam Questions

READ AND HIGHLIGHT THE QUESTION

BE SPECIFIC: Talk exact features

On questions about **designing backup systems**…

- **Always** mention **testing** the **backup procedure** and **training staff** to **cope** - ***you need these to get full marks***

On questions about **the Human Computer Interface**…

- **Always** talk about features that will **reduce** the **chance** of **errors** in the **input** ***(i.e. same layout screen as form, etc.)***

On questions about **flat file databases**…

- **Always mention redundancy**. Usually this will get you the **marks** for **one point**, its **consequences** can make up **another point** ***(i.e. discrepancies between files)***