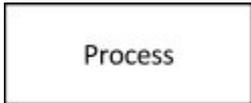
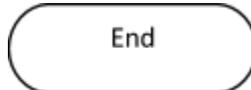
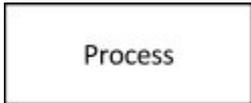
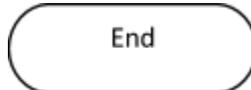
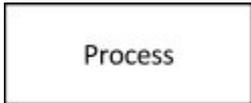
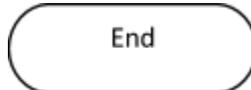


# Computer Science Revision Notes

Specification Area	Notes
<ul style="list-style-type: none"> <li><b>Data and Information</b></li> </ul>	<p><b>Data</b> can consist of facts or statistics out of the context and used for analysis. It can be numbers, characters, symbols or images that are processed by a computer.</p> <p><b>Information</b> can be a sentence of words, a sequence of number or a series of images that have been put into a context, which is what gives the data meaning.</p>
<ul style="list-style-type: none"> <li><b>Constants and Variables</b></li> </ul>	<p>Variables are used in computer programming to store specific values within a program. They are assigned both a data type and a value. Some values are stored permanently; the data is <b>constant</b>. Other things are always changing; the data is <b>variable</b>.</p> <p><b>Variables</b> are data entities whose values can be altered.</p> <p><b>Constants</b> are data entities whose values cannot be altered.</p>
<ul style="list-style-type: none"> <li><b>Understanding different data types</b></li> </ul>	<p>A data type determined the actions that can be performed on data when it is held within a program or field of a database or a spreadsheet.</p> <p>The most common data types are called <b>primitive data types</b>.</p> <p>The different primitive data types are explained below:</p> <ul style="list-style-type: none"> <li>● <b>Integer:</b> An integer data type deals with whole numbers. A member of the set of positive whole numbers, negative whole numbers and zero.</li> <li>● <b>Real:</b> A real data type contains numeric data in a decimal</li> <li>● <b>Boolean:</b> The Boolean data type represents values of true/false or yes/no. The primitive data type of a Boolean is logical. There are arithmetic tests that can be used to create Boolean values. They are &lt;, &lt;=, ==, !=, &gt;= and &gt;</li> <li>● <b>String:</b> A string is a data type that can contain a set of characters, numbers and spaces.</li> <li>● <b>Character:</b> A character type may contain a <b>single</b> letter, digit, punctuation mark, symbol, formatting code, control code, or some other specialised code.</li> </ul>
<ul style="list-style-type: none"> <li><b>Data structures</b></li> </ul>	<p><b>What is a data structure?</b></p> <p>A data structure is a specialized format for organizing and storing data. General data structure types include the array, the file, the record, the table, the tree, and so on. Any data structure is designed to organize data to suit a specific purpose so that it can be accessed and worked with in appropriate ways.</p> <p><b>Examples of data structures:</b></p> <ul style="list-style-type: none"> <li>● <b>Array:</b> An array is a series of memory locations which holds a series of items of data. All data in an array must be of the same data type.</li> <li>● <b>List:</b> A list is a series of memory location which can hold a series of items of data. All data in an array can be of different data types.</li> </ul> <p><b>Explain why using data structures make coding simpler:</b></p> <p>A variable holds a single item of data. There may be a situation where lots of variables are needed to hold similar and related data. In this situation, using a data structure can simplify a program by storing all related data under one</p>

	<p>name. This means that a program can be written to search through the data structure much more quickly than having to write a new line of code for every variable. This reduces the complexity and length of the program which makes it easier to find and debug errors.</p>								
<ul style="list-style-type: none"> <li>● <b>Program Flow Control</b></li> </ul>	<p><b>Program flow control</b> can be defined simply as the order in which a program is executed</p> <p>Advantage of using it when designing programs</p> <ul style="list-style-type: none"> <li>● It decreases the complexity of the task at hand, as everything is completed in a logical manner and so the program should take less time to code</li> <li>● This logical structure also means that the code has increased clarity and so faults can be recognised quicker and more easily</li> <li>● It also means that many programmer can work on one program, as the structure has already been planned and is fairly standards. This also means in an increase in productivity.</li> </ul> <p><b>Representing structure:</b></p> <p>Flow charts can be used to plan programs and represent structure The four main symbols used in making flow charts are shown below:</p> <table border="1" data-bbox="523 954 1406 1536"> <tr> <td data-bbox="523 954 943 1099">  </td> <td data-bbox="943 954 1406 1099"> <p>This shows the start of a process and has only one output</p> </td> </tr> <tr> <td data-bbox="523 1099 943 1245">  </td> <td data-bbox="943 1099 1406 1245"> <p>This represents an operation carried out. This blocks must have exactly one input and one output.</p> </td> </tr> <tr> <td data-bbox="523 1245 943 1391">  </td> <td data-bbox="943 1245 1406 1391"> <p>This represents a question with only two answers so it always makes a Boolean choice. It has one input and two outputs.</p> </td> </tr> <tr> <td data-bbox="523 1391 943 1536">  </td> <td data-bbox="943 1391 1406 1536"> <p>This is the opposite of the 'start' symbol. It only has one input</p> </td> </tr> </table> <p>On the other hand, <b>algorithms</b> can also be used. An algorithm is a series of instructions that solves a problem in a finite number of steps that always ends.</p> <p>The basic building blocks of coded solution are sequencing, selection and iteration. They are explained below:</p> <ul style="list-style-type: none"> <li>● A <b>sequence</b> is a set of instructions acting in order, meaning that each action follows the previous action.</li> <li>● A <b>selection</b> control structure involves a choice. Usually, a program's execution path is determined by statements.</li> <li>● An <b>iteration</b> control structure involves repeating actions. It involves a statement that is executed in a loop until its criteria are met.</li> </ul>		<p>This shows the start of a process and has only one output</p>		<p>This represents an operation carried out. This blocks must have exactly one input and one output.</p>		<p>This represents a question with only two answers so it always makes a Boolean choice. It has one input and two outputs.</p>		<p>This is the opposite of the 'start' symbol. It only has one input</p>
	<p>This shows the start of a process and has only one output</p>								
	<p>This represents an operation carried out. This blocks must have exactly one input and one output.</p>								
	<p>This represents a question with only two answers so it always makes a Boolean choice. It has one input and two outputs.</p>								
	<p>This is the opposite of the 'start' symbol. It only has one input</p>								

<ul style="list-style-type: none"> <li>● <b>Procedures and functions</b></li> </ul>	<p>A <b>procedure</b> is a block of code that performs a task without returning a value. A <b>function</b> is very similar to a procedure except that it returns a value. There are two types of functions;</p> <ul style="list-style-type: none"> <li>● <b>User-defined functions</b> are created by the user for a program.</li> <li>● <b>Built-in functions</b> are part of the programming language. Built-in functions are very useful and save a lot of effort in writing code to perform common tasks.</li> </ul> <p>Procedure and functions are subroutines that feature in all programming languages. They are named blocks of code that can be used and re-used to perform specific tasks. Their purpose is to allow the repetition of certain sections of a program or calculation many times.</p> <p><b>Reasons why programmers use functions and procedures:</b></p> <ul style="list-style-type: none"> <li>● To make code easier to read</li> <li>● To allow for code reuse/sharing</li> <li>● To reduce programmer error</li> <li>● To improve code maintenance/easier to update code.</li> </ul> <p>A <b>parameter</b> is used in a subroutine to refer to one of the pieces of data provided as input to the subroutine. It therefore allows the passing of values to the procedure or function for use inside it.</p> <p>A <b>return value</b> is the information that comes back from the actions performed from the subroutine.</p>
<ul style="list-style-type: none"> <li>● <b>Scope</b></li> </ul>	<p><b>Scope</b> is the area in the code where a variable can only be accessed and changed. Trying to access a variable in other parts of the code that do not fall in this area will not work. There are two types of scopes. <b>Local</b> scope which is within a function, and <b>global</b> scope which is anywhere in the program. The different types of scopes mean there are two different types of variables. These are <b>local variables</b> and <b>global variables</b>.</p> <p>The scope of a local variable is the procedure in which it is declared. The scope of a global variable is the whole program.</p>
<ul style="list-style-type: none"> <li>● <b>Error Handling</b></li> </ul>	<p><b>Types of Errors:</b></p> <p>There are three types of error that computer programmers encounter when writing software. They are <b>syntax errors</b>, <b>run-time errors</b> and <b>logic errors</b>.</p> <ul style="list-style-type: none"> <li>● <b>Syntax error:</b> They occur when the programmer fails to obey one of the grammar rules of the programming language that they are using.</li> <li>● <b>Run-time error:</b> They occur whenever the program instructs the computer to carry out an operation that is either not designed to do or slow to do.</li> <li>● <b>Logic error:</b> They occur when the program runs successfully but it does not behave in the manner it was designed to.</li> </ul> <p><b>Why are logical errors often the most difficult to find?</b></p> <p>This is because the program appears to run normally and because it is not obvious where the error has occurred.</p>

	<p><b>Error Handling:</b> You must understand that some errors can be detected and corrected during the coding stage. You must also understand that some errors will occur during the execution of the code.</p> <p>The process of testing a program for errors during the execution of code is a cyclic activity called <b>debugging</b>.</p> <p>Debugging can also be enhanced with the aid of tools. Some of these tools are stated below:</p> <ul style="list-style-type: none"> <li>● Watch: <ul style="list-style-type: none"> <li>○ As the name implies, this essentially ‘watches’ any variable specified by the programmer</li> <li>○ They show its contents relative to the line that the program is currently executing</li> </ul> </li> <li>● Breakpoints: <ul style="list-style-type: none"> <li>○ These can be added to the code by the programmer which will halt the program’s execution when it reaches that point</li> <li>○ They allow the program to test the code at those points</li> </ul> </li> <li>● Step: <ul style="list-style-type: none"> <li>○ Allows the programmer to continue the program’s execution on a line-by-line basis, to see what is changed on each line</li> </ul> </li> <li>● Use of an Integrated Development Environment/IDE</li> <li>● Syntax colouring</li> <li>● Code completion</li> <li>● Automated/Integrated testing</li> <li>● Compiler/interpreter</li> <li>●</li> </ul> <p>Some errors can be detected before even making the code. This is called <b>dry-run testing</b>. It is usually carried out on the algorithm which is written in pseudo-code or as part of a flowchart. The main advantage to dry-run testing is that it enables programmers to spot errors even before they start writing code.</p> <p><b>Trace Tables:</b> A trace table is used to test algorithms to see if any logic errors occur while the algorithm is being processed.</p>
<ul style="list-style-type: none"> <li>● <b>Systems</b></li> </ul>	<p>You must know what a <b>computer system</b> is. It is hardware and software working together to create a working solution. In its very basic form, a computer can be looked as nothing more than an input, a process, some storage, and an output.</p> <p><b>Hardware</b> is the machines, wiring, and other physical components of a computer or other electronic system.</p> <p><b>Software</b> is the programs and other operating information used by a computer.</p> <p>Computer system are integral to our modern world as:</p> <ul style="list-style-type: none"> <li>● They allow people to contact and work with those all around the world</li> <li>● They allow people to access their data securely from anywhere</li> </ul>

	<ul style="list-style-type: none"> <li>● They improve quality of life, e.g. those without limbs can have then replaced with bionic</li> <li>● They help save lives, e.g. emergency response teams can coordinate rescues quicker</li> </ul> <p><b>Reliability</b> in computer systems is vitally important. High reliability is when a computer system can be depended on to work almost all of the time. Robustness is also vital. <b>Robustness</b> is the ability of a computer system to cope with execution errors or abnormalities.</p> <p>Both are important as high reliability and robustness mean that the computer system can be depended on at all times without any chance of failure</p> <p>If a system is not reliable or robust then the user may not be able to use the program when required, which, depending on the situation, could have severe consequences.</p> <p>One way computer systems are made reliable is the duplication of each component in a system or the whole system itself, so that if there is an error, there is always a backup to replace them in case they stop working.</p>
<ul style="list-style-type: none"> <li>● <b>Hardware</b></li> </ul>	<p><b>Hardware</b> is the machines, wiring, and other physical components of a computer or other electronic system.</p> <p>The fundamental pieces of hardware required to make a functioning computer system are stated below:</p> <ul style="list-style-type: none"> <li>● CPU: The 'central processing unit' undertakes instructions it receives from programs in what is called a cycle.</li> <li>● Motherboard: This links the CPU to the memory and other hardware.</li> <li>● Graphics Card: They contain a processor like the CPU and can be used to process the graphic portion of the processing load.</li> <li>● Hard Drive: This store the majority of your data. The hard drive unit houses the hard disk, where the data is physically stored.</li> <li>● ROM (read only memory): ROM is non-volatile. The computer's BIOS/ Initial instructions/bootstrapping instructions are stored in the non-volatile memory in a computer's main memory.</li> <li>● RAM (random access memory): RAM is volatile. When a program is opened, it get loaded from the hard drive into the RAM, because reading data from RAM is much quicker, making the program's use quicker.</li> </ul> <p>It is important that you know how developments in different hardware technologies (including memory and processor) are leading to exciting innovative products being created, like in the mobile and gaming industries. Generally, the size of the hardware is decreasing, allowing them to fit inside smaller cases. This means that phones, tablets and games consoles can use hardware that is almost as powerful as home computers. Developments in hardware have made smart phones possible include:</p> <ul style="list-style-type: none"> <li>● More power efficient processors</li> <li>● Smaller form memory</li> <li>● Solid state memory</li> <li>● Affordable touchscreens</li> </ul> <p>In all computer systems there are also <b>input</b> and <b>output</b> devices. These are essential for computer systems to work well with a user.</p>

- **CPU (central processing unit) and Motherboard**

The CPU undertakes instructions it receives from programs in what is called a cycle. The CPU has a number of cores. The speed of a CPU is measured in how many cycles it can perform in a second. This is regarded as clock speed. The name given to one cycle per second is a Hertz. A CPU's control unit performs the following instructions:

- Fetch instructions
- Decode instructions
- Execute instructions
- Store results.

This is known as the processor cycle.

The characteristics that can improve the performance of CPUs are shown below:

- the cache size
- the type of cache memory
- the number of cores/processors
- the processing clock speed
- the bus width
- the word size
- the architecture of the processor/CPU

The performance of a CPU can be explained by the following factors as well as others:

- Clock speed
  - This is the speed at which the a microprocessor (meaning contained on one chip) executes instructions
  - A higher clock speed means a process will be completed quicker
  - However increased clock speed means increased heat produced and more power consumes
- Number of cores
  - A computer can have CPUs with multiple cores.
  - When you have multiple CPU cores on a single chip, they are called multi-core processors
  - The multiple cores can run multiple instructions at any given time which is they are effective when the user is multitasking.
  - However, software must specifically written to cater for multiple cores so an increase in speed may not be obvious for some applications.
- Cache
  - Cache stores recently used data so that it can be quickly accessed at a later time
  - Processor caches are small data stores which contain regularly used instruction that can be accessed quickly by the CPU when required
  - When the CPU finds stored data in the cache it is called a 'cache hit', so a larger amount of cache means that these are more like, so the processor will operate quicker.

	<p>The main differences between mobile and non-mobile processors are described below:</p> <ul style="list-style-type: none"> <li>● Clock Speed: Lower clock speeds mean the amount of heat produced is less. Therefore phones have lower clock speed processors.</li> <li>● Due to less space in portable devices, other components are integrated with the processor, like the wireless networking components.</li> <li>● Mobile processors are also optimised for low power consumptions.</li> </ul> <p>A motherboard links the CPU and PU's to the memory and other hardware. The CPU uses the motherboard to receive and send signals to components, such as the hard drives (storage). Motherboards have buses (circuits) that connect one part of the motherboard to another. The speed of the bus is measured in Megahertz. The more data the motherboard bus can handle at any one time, the faster the system.</p>
<ul style="list-style-type: none"> <li>● <b>Memory</b></li> </ul>	<p>There are two main categories of chip-based memory: volatile memory and non-volatile memory. Data and instructions are stored in memory and processed by the CPU in order to produce a desired result or output.</p> <p><b>Volatile Memory:</b> volatile memory is computer memory that requires a power supply in order to maintain stored data. The moment that the power is turned off the information is lost. The RAM (random access memory) in your computer is a type of volatile memory. The volatile memory cannot store data when the computer device is turned off.</p> <p><b>Non-volatile Memory:</b> Non-volatile memory is computer memory that retains its information whether the power being supplied to it is turned off or on. An example of non-volatile memory include read only memory (ROM) and flash memory, as well as most types of magnetic hard disks and optical disks.</p> <p><b>ROM (read only memory):</b> ROM is non-volatile. The computer's BIOS/ Initial instructions/bootstrapping instructions are stored in the non-volatile memory in a computer's main memory.</p> <p><b>RAM (random access memory):</b> RAM is volatile. When a program is opened, it get loaded from the hard drive into the RAM, because reading data from RAM is much quicker, making the program's use quicker.</p> <p><b>Why is it important for computers to have a sufficient amount of RAM?</b></p> <ul style="list-style-type: none"> <li>● A large amount of RAM enable more instructions to be loaded</li> <li>● From the secondary storage into RAM, so they can be executed by the processor.</li> </ul> <p><b>Cache Memory:</b> Cache memory is the memory closest to the CPU. All the recent instructions are stored in cache memory as it is very fast. A cache memory often has a small access time, whereas the access time from the main memory is larger. Cache memory is very expensive and hence is limited in capacity.</p>

	<p><b>Explain why having cache memory can improve the performance of the central processing unit (CPU):</b></p> <ul style="list-style-type: none"> <li>● Frequently used data/instructions are stored in the cache</li> <li>● Meaning they don't have to be fetched from main memory.</li> <li>● Therefore data/instructions stored in the cache memory can be accessed faster than data stored in the main memory.</li> </ul> <p><b>Virtual Memory:</b> A section of volatile memory on a storage drive. It is used when RAM is running low, some of the HDD is used as RAM. It is much slower than RAM as it takes up more processing power to move the data around.</p>
<ul style="list-style-type: none"> <li>● <b>Secondary Storage</b></li> </ul>	<p>Secondary storage is non-volatile memory that is used to store large amount of data permanently. It is also known as auxiliary/external memory</p> <p>Example are:</p> <ul style="list-style-type: none"> <li>● Flash memory <ul style="list-style-type: none"> <li>○ Solid State Memory</li> <li>○ USB Memory sticks</li> </ul> </li> <li>● Optical media <ul style="list-style-type: none"> <li>○ CDs</li> <li>○ DVDs</li> </ul> </li> <li>● Magnetic media <ul style="list-style-type: none"> <li>○ Hard disk drives</li> <li>○ Floppy disks</li> </ul> </li> </ul> <p><b>How do they store data?</b></p> <ul style="list-style-type: none"> <li>● <b>Optical Media:</b> Digital optical discs such as CDs and DVDs store binary data as microscopic variations in the height of the disc's surface, known as pits and lands. Bits are encoded as transitions between raised lands and etched pits, which are lined up in a spiral. To read the data an infrared laser is beamed through the CD's material. The light that bounces off the mirror-like reflective backing is then measured. Because pits and lands pass different amounts of light, an opto-electric sensor can read the stamped pattern by detecting fluctuations in the reflected beam as the laser scans the groove. These measurements can then be translated back into the original ones and zeros.</li> <li>● <b>Magnetic Media:</b> Magnetic storage media and devices store data in the form of tiny magnetised dots. These dots are created, read and erased using magnetic fields created by very tiny electromagnets. In the case of magnetic tape the dots are arranged along the length of a long plastic strip which has been coated with a magnetisable layer (audio and video tapes use a similar technology). In the case of magnetic discs (e.g. floppy disc or hard-drive), the dots are arranged in circles on the surface of a plastic, metal or glass disc that has a magnetisable coating.</li> <li>● <b>Solid State:</b> Flash memory is solid. There are no moving parts. A SSD uses microscopic electronic switches to store data. Millions of transistors are</li> </ul>

	<p>either on or off and remain so as long as electricity is applied to the system. SSDs use switches called floating gate transistors. These create a special cage (the gate) that traps the electron flow within it. This gate effect continues whether there is electricity in the system or not, and so storage remains when the power is removed.</p> <p><b><u>Advantages of Solid state media over Magnetic Media:</u></b></p> <ul style="list-style-type: none"> <li>● <b>Difference:</b> No mechanical parts in solid state media//Magnetic media has mechanical parts. <b>Explanation:</b> Magnetic media are often unsuitable for mobile use because the mechanical parts cannot function during movement // mechanical parts are less robust during movement.</li> <li>● <b>Difference:</b> Speed of read access higher in solid state drives. <b>Explanation:</b> Data can often be read more quickly from solid state media than magnetic media.</li> <li>● <b>Difference:</b> Solid state media can be more compact than magnetic media. <b>Explanation:</b> The smaller size enables better mobility;</li> <li>● <b>Reason:</b> The battery will last longer <b>Explanation:</b> Solid state media uses less power</li> <li>● <b>Difference:</b> Less heat generated when using solid state <b>Explanation:</b> Utilising the power more efficiently//allows for more miniaturisation.</li> <li>● <b>Difference:</b> Solid state is silent <b>Explanation:</b> Makes it more attractive to use.</li> </ul>
<ul style="list-style-type: none"> <li>● <b>Algorithms</b></li> </ul>	<p>Explain what an algorithm is:</p> <ul style="list-style-type: none"> <li>● A series of instructions</li> <li>● That solves a problem</li> <li>● In a finite number of steps that always ends</li> </ul>
<ul style="list-style-type: none"> <li>● <b>Data Representation</b></li> </ul>	<p>Computers use the binary alphabet to represent all data and instructions. Binary code consists of two digits: 0 and 1. They symbolise voltage with 0 being off and 1 being on. Binary code could represent different types of data such as text, image, sound, integer, date and real number.</p> <p>A bit has a value of 1 or 0.  A nibble is four bits  A byte is eight bits.  A kilobyte (KB) is 1024 bytes.  A megabyte (MB) is 1024 kilobytes.  A gigabytes (GB) is 1024 megabytes.  A terabyte (TB) is 1024 gigabytes.</p> <p><b>You must know how to use binary to represent whole numbers up to 2<sup>5</sup></b></p>

Always remember to draw this table when converting binary to denary or denary to binary.

128	64	32	16	8	4	2	1

**You must know how to convert between binary, denary and hexadecimal.**

When converting denary to Hexadecimal, use the following table.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

**The stages of converting sound to digital form/binary:**

- Computer cannot work with analogue data so it must be converted to digital data.
- Microphone picks up sound waves.
- Converted to an electrical analogue signal.
- Value read at specific point and rounded to a level.
- The points which sound is read and stored defines the sample rate. The higher the sample rate the better the quality but the larger the sound file.
- Binary representation of level stored.

**The stages of using binary to represent bitmap images:**

- The image is represented as a grid or sequence of pixels.
- Each pixel is represented by one bit.
- White is represented by a 0
- Black is represented by a 1
- White and black are represented using different bit patterns.
- Same bit pattern used for black and white.
- Metadata about the image is also stored.

**ASCII:**

Characters are represented in binary using **ASCII** or other character conversion systems. ASCII has some limitations. The total number of characters that can be encoded in ASCII is 128 Characters.

**Explain the possible limitations of using ASCII character set for global communication:**

- ASCII is a 7-bit character set so can include at most 128 different characters
- These 128 characters represent mainly just the Latin alphabet
- And so this means that the characters of many other alphabets/languages cannot be represented.

**Advantages of using hexadecimal over binary:**

	<ul style="list-style-type: none"> <li>● Hexadecimal is easier (for humans) to read than binary</li> <li>● Hexadecimal is easier to convert to binary than denary.</li> <li>● Numbers are displayed in a more compact way in hexadecimal than in binary.</li> <li>● It is quicker to type in hexadecimal numbers than binary numbers</li> <li>● Less memory is also used to store hexadecimal than binary.</li> <li>● Easier to remember as they are shorter.</li> </ul>
<ul style="list-style-type: none"> <li>● <b>Software Development Cycle</b></li> </ul>	<p>The software development life cycle is a structured procedure used for developing new software products:</p> <ul style="list-style-type: none"> <li>● Step 1 – Requirement Analysis <ul style="list-style-type: none"> <li>○ At this stage the programmers, designers and the clients discuss the needs of the user so it is clear what is required</li> </ul> </li> <li>● Step 2 – Design <ul style="list-style-type: none"> <li>○ Here the solution is planned, using the specification discussed in the previous section</li> </ul> </li> <li>● Step 3 – Implementation <ul style="list-style-type: none"> <li>○ In this section the designs are turned in code</li> <li>○ This is usually the longest phase</li> </ul> </li> <li>● Step 4 – Testing <ul style="list-style-type: none"> <li>○ During this phase the code written during the implementation section is tested against the specification, to ensure the code written really provides a solution to the problem</li> </ul> </li> <li>● Step 5 – Evolution <ul style="list-style-type: none"> <li>○ These are changes/upgrade that the software goes through after it is delivered to the client</li> <li>○ This stage can only occur if the programmers bear this in mind during the design and the implementation stages, as the software needs to be coded to allow future changes to be made</li> </ul> </li> </ul> <p>There are several lifecycle models that can be used. Some of these are cyclical, waterfall, and the spiral model.</p> <p><b>Cyclical Model:</b> This model is explained above and it is more of a generic model.</p> <ul style="list-style-type: none"> <li>● <b>Advantages:</b> <ul style="list-style-type: none"> <li>○ Clear defined stages.</li> <li>○ Simple to understand and use.</li> <li>○ Well understood milestones.</li> <li>○ Easy to arrange tasks.</li> </ul> </li> <li>● <b>Disadvantages:</b> <ul style="list-style-type: none"> <li>○ It's difficult to measure progress within stages.</li> <li>○ Cannot accommodate changing requirements.</li> <li>○ No working software is produced until late in the cycles</li> <li>○ Risk and uncertainty is high with this process model</li> </ul> </li> </ul> <p><b>Waterfall Model:</b> In a waterfall model, each phase must be completed in its entirety before the next phase can begin. At the end of each phase, a review takes</p>

	<p>place to determine if the project is on the right path and whether or not to continue or discard the project. Unlike what I mentioned in the general model, phases do not overlap in a waterfall model.</p> <ul style="list-style-type: none"> <li>● <b>Advantages:</b> <ul style="list-style-type: none"> <li>○ Simple and easy to use.</li> <li>○ Easy to manage due to the rigidity of the model – each phase has specific deliverables and a review process.</li> <li>○ Phases are processed and completed one at a time.</li> <li>○ Works well for smaller projects where requirements are very well understood.</li> </ul> </li> <li>● <b>Disadvantages:</b> <ul style="list-style-type: none"> <li>○ Adjusting scope during the life cycle can kill a project</li> <li>○ No working software is produced until late during the life cycle.</li> <li>○ High amounts of risk and uncertainty.</li> <li>○ Poor model for complex, object-oriented and long projects.</li> <li>○ Can be difficult to go back a step if needed.</li> <li>○ Poor model where requirements are at a moderate to high risk of changing.</li> </ul> </li> </ul> <p><b>Spiral Model:</b> The spiral model has four phases: Planning, Risk Analysis, Engineering and Evaluation. A software project repeatedly passes through these phases in iterations (called Spirals in this model). The baseline spiral, starting in the planning phase, requirements are gathered and risk is assessed. Each subsequent spirals builds on the baseline spiral.</p> <ul style="list-style-type: none"> <li>● <b>Advantages:</b> <ul style="list-style-type: none"> <li>○ High amount of risk analysis</li> <li>○ Good for large and mission-critical projects.</li> <li>○ Software is produced early in the software life cycle.</li> <li>○ More client consultation</li> <li>○ Ability to return and fix problems.</li> </ul> </li> <li>● <b>Disadvantages:</b> <ul style="list-style-type: none"> <li>○ Can be a costly model to use.</li> <li>○ Risk analysis requires highly specific expertise.</li> <li>○ Project’s success is highly dependent on the risk analysis phase.</li> <li>○ Doesn’t work well for smaller projects.</li> </ul> </li> </ul>
<ul style="list-style-type: none"> <li>● <b>Prototyping</b></li> </ul>	<p>Prototyping offers people who have an interest in a system the capability of evaluating the proposed designs, rather than having to interpret those designs on some other means such as descriptions.</p> <p><b>Advantages of prototyping:</b></p> <ul style="list-style-type: none"> <li>● It allows the customer and the developer to compare if the developed software matches the software specification according to which the program should have been built.</li> <li>● It can offer improvements to the quality and definition of customer</li> </ul>

	<p>requirements and improve the specifications provided to developers.</p> <ul style="list-style-type: none"> <li>● Making a prototype first often results in faster and less expensive software and less wasted time.</li> <li>● The existence of a prototype enables detailed examination by users, and this prevents the misunderstandings and miscommunications that can occur when each side thinks the other understands what they said.</li> </ul> <p><b>Disadvantages of prototyping:</b></p> <ul style="list-style-type: none"> <li>● Prototypes are limited in functionality and may not scale well if they are used as the basis for a final product.</li> <li>● Users can become attached to functionality that was included in a prototype for consideration and then removed from the specification for the final system; this can lead to conflict.</li> <li>● Prototyping is supposed to be carried out quickly. Some developers may try to produce complex prototypes.</li> <li>● Prototyping is expensive.</li> </ul>
<ul style="list-style-type: none"> <li>● <b>Application Testing</b></li> </ul>	<p><b>Application testing</b> is using the application in controlled condition to see how it works</p> <ul style="list-style-type: none"> <li>● Rigorous testing is essential for coded solutions as it determines whether or not the developers have met the specification outlined at the start of the software development life cycle</li> <li>● It also ensure that the end user's (those who will operate the software) experience is not affected by bugs and errors, which make the evolution phase easier as less has to be done</li> </ul> <p><b>Test plans</b></p> <ul style="list-style-type: none"> <li>● These are documents created before the testing phase of the software developmental life cycle outlining what tests will occur and how they will be conducted</li> <li>● They should show the methods used and the anticipated results</li> <li>● The details outlined in these plans must be quantifiable and measurable</li> </ul> <p><b>Types of tests:</b></p> <ul style="list-style-type: none"> <li>● <b>Unit Testing:</b> Unit tests or modular tests are performed by the programmer as they are developing the program. They test the correctness of small blocks of code in isolation. <ul style="list-style-type: none"> <li>○ Advantages of unit testing: <ul style="list-style-type: none"> <li>▪ Problems are discovered early.</li> <li>▪ Integration testing is easier.</li> <li>▪ Acts as documentation of the system.</li> <li>▪ Extremely accurate unit test coverage.</li> <li>▪ The unit test itself is used to verify the design.</li> </ul> </li> <li>○ Disadvantages of unit testing: <ul style="list-style-type: none"> <li>▪ Time consuming.</li> <li>▪ Does not show absence of errors.</li> <li>▪ Hard to set up realistic, useful tests.</li> </ul> </li> </ul> </li> </ul>

	<ul style="list-style-type: none"> <li>▪ Integration errors or system errors can be missed.</li> <li>▪ May be necessary to use a version control system.</li> <li>▪ Need to have a review process for test case failures.</li> <li>▪ Test program cannot be run in actual deployment environment.</li> <li>▪ Test code is likely to be at least as buggy as the code it is testing.</li> </ul> <ul style="list-style-type: none"> <li>● <b>Integration Testing:</b> Integration testing is performed when all of the different parts of code, such as functions or modules, are complete. This tests whether the units of code work together correctly. <ul style="list-style-type: none"> <li>○ Advantages of integration testing: <ul style="list-style-type: none"> <li>▪</li> </ul> </li> <li>○ Disadvantages of integration testing: <ul style="list-style-type: none"> <li>▪</li> </ul> </li> </ul> </li> <li>● <b>System Testing:</b> System testing takes place after the code has been individually tested and is done without having to have any knowledge of the code itself. This tests that the entire system functions correctly</li> <li>● <b>Non-functional Testing:</b> This tends to reflect the quality of the product, particularly the suitability of the application from the point of view of the client. A load test investigates software behaviour during increasing system loads and a performance test is carried out to investigate the processing speed and response time for specific scenarios.</li> <li>● <b>Functional testing:</b> This is testing in normal conditions. This tests to ensure that the software is in line with all of the specifications outlined by the client. Usually, the software should have been built for this type of data.</li> </ul>
<ul style="list-style-type: none"> <li>● <b>Networking</b></li> </ul>	<p>What is a <b>computer network</b>:</p> <ul style="list-style-type: none"> <li>● Two or more computers, that have been connected together and that can communicate with one another.</li> </ul> <p><b>Advantages of using a computer network:</b></p> <ul style="list-style-type: none"> <li>● Users can share software stored in a main system.</li> <li>● Site (network) licences for software are less expensive than several standalone licences.</li> <li>● Files can be shared between users.</li> <li>● Users can communicate via email, instant messenger and VoIP.</li> <li>● Security is of a high standard.</li> <li>● It is much more straightforward to back up data to a file server.</li> <li>● Data can be transmitted to remote areas.</li> <li>● Users can share common peripheral resources such as printers, fax machines, modems etc. and therefore saving money.</li> </ul> <p><b>Disadvantages of using a computer network:</b></p> <ul style="list-style-type: none"> <li>● Cabling and file servers can be costly.</li> </ul>

- Management is complicated, requiring training and a specialist network manager.
- If a file server breaks down, the files contained on it become inaccessible.
- If a virus gets into the system, it can easily spread to other computers.
- There is a risk of hacking, particularly in wide area networks.
- Rigorous security measures are required.

Networks can be organised in different formats. These formats are called network topologies.

**Bus topology:** In this network configuration all of the clients, servers and resources are connected to one cable (the bus). When a device communicates, the message is transmitted down the bus that all devices receive, but only the intended recipient accepts and processes the message

- **Advantages of the bus topology:**
  - It is easy and cheap to install as a consequence of requiring only a small quantity of cable.
  - It is suitable for small networks.
- **Disadvantages of bus topology:**
  - The cable length is limited. This restricts the number of devices that can be connected to the network.
  - As more devices are connected, the performance of the network becomes slower as a consequence of data collisions.
  - All workstations see all data – so there is a security risk.

**Ring topology:** In this topology, each device is connected to two other devices which forms a ring for the messages to travel around. When a message is sent in a single direction it is received by each device until the intended recipient receives it.

- **Advantages of the Ring topology:**
  - Data is transferred quickly as data only flows in a single direction so there are no data collisions
- **Disadvantages of Ring topology:**
  - Workstations often see data not intended for them – so there is a security/privacy risk
  - If any of the cable fails or a device is faulty then the whole network will fail.
  - Alterations or maintenance to the network nodes can impact the performance of the while network.

**Star topology:** Most home network use this topology as it has a central connection point that could be a hub, switch or router.

- **Advantages of the Star topology:**
  - They are reliable as if one cable or device fails then the other devices will be unaffected
  - They are high performing as no data collision occur.
  - Changes to one client do not affect the others.
- **Disadvantages of Star topology:**

	<ul style="list-style-type: none"> <li>○ Expensive to install as lots of cabling is required and extra hardware (the central device)</li> <li>○ If the central device fail, then so will the whole network.</li> </ul> <p><b>Basic components of a computer network:</b></p> <ul style="list-style-type: none"> <li>● Servers are powerful computers that provide services to other computers on the network.</li> <li>● Clients are computers that use the services that a server provides.</li> <li>● Communication media like cables or a wireless signal.</li> <li>● Peripheral devices include printers and fax machines etc.</li> <li>● A user is a person that uses the client machines.</li> <li>● The protocols of a network are rules for the network communication.</li> <li>● A network adapter is a circuit board that can send and receive data.</li> </ul>														
<ul style="list-style-type: none"> <li>● <b>Client server</b></li> </ul>	<p>The <b>client-server model</b> is the structure of a computer network in which many clients request and receive services from a centralised server.</p> <p>For computers to work together they have to use protocols. A <b>protocol</b> is a set of rules which governs the transfer of data between computers.</p> <p>When two computers first connect in a network they use a handshake. <b>Handshaking</b> establishes which protocols to use and controls the flow of data between the two or more connected computers.</p> <p><b>Handshaking process:</b></p> <ol style="list-style-type: none"> <li>1. A request to connect is sent.</li> <li>2. A connection is granted</li> <li>3. Protocol parameters are agreed.</li> </ol> <p><b>Some of the differences between a standalone application and a client-server application are:</b></p> <table border="1" data-bbox="525 1377 1506 1648"> <thead> <tr> <th style="background-color: #e91e63; color: white;">Stand-Alone Applications:</th> <th style="background-color: #e91e63; color: white;">Client-Server Applications:</th> </tr> </thead> <tbody> <tr> <td>All data stored locally (disk or main memory)</td> <td>Data must be passed between the client and server</td> </tr> <tr> <td>No network connection to manage</td> <td>Must manage a network link and the associated sessions</td> </tr> <tr> <td>Single user</td> <td>Multiple clients per server so must manage a request queue</td> </tr> <tr> <td>Used as needed, can be closed in between uses</td> <td>Must be available all the time</td> </tr> <tr> <td>Modular structure less important</td> <td>Must be designed and created in a modular fashion</td> </tr> <tr> <td>Local access rights on host PC is sufficient security</td> <td>Clients have to be authenticated before data is sent to them</td> </tr> </tbody> </table>	Stand-Alone Applications:	Client-Server Applications:	All data stored locally (disk or main memory)	Data must be passed between the client and server	No network connection to manage	Must manage a network link and the associated sessions	Single user	Multiple clients per server so must manage a request queue	Used as needed, can be closed in between uses	Must be available all the time	Modular structure less important	Must be designed and created in a modular fashion	Local access rights on host PC is sufficient security	Clients have to be authenticated before data is sent to them
Stand-Alone Applications:	Client-Server Applications:														
All data stored locally (disk or main memory)	Data must be passed between the client and server														
No network connection to manage	Must manage a network link and the associated sessions														
Single user	Multiple clients per server so must manage a request queue														
Used as needed, can be closed in between uses	Must be available all the time														
Modular structure less important	Must be designed and created in a modular fashion														
Local access rights on host PC is sufficient security	Clients have to be authenticated before data is sent to them														
<ul style="list-style-type: none"> <li>● <b>Web Application concepts</b></li> </ul>	<p>Server-side programming is writing code that runs on the server, using languages supported by the server (such as Java, PHP, C#; it is possible to write code that executes on the server-side in JavaScript). Client-side programming is writing code that will run on the client, and is done in languages that can be executed by the browser, such as JavaScript</p> <p><b>Server-side programming:</b></p> <ul style="list-style-type: none"> <li>● Retrieving data from a database.</li> <li>● Populating a page with the latest details.</li> </ul>														

	<ul style="list-style-type: none"> <li>● Checking username and password.</li> <li>● Some forms of validation.</li> <li>● Counting how many people have visited a site.</li> </ul> <p><b>Client-side programming:</b></p> <ul style="list-style-type: none"> <li>● Validation rules</li> <li>● Flash games</li> <li>● Cookies</li> <li>● Advertising</li> <li>● Responsive design</li> </ul>
<ul style="list-style-type: none"> <li>● <b>Use of external code sources</b></li> </ul>	<p>External code sources are sources that are not designed specifically by a user, but has a generic purpose. These can be free or even cost money.</p> <p><b>The advantages of using external code sources:</b></p> <ul style="list-style-type: none"> <li>● Saves them programming time</li> <li>● Allows more professional looking input fields</li> <li>● Allows access to advanced validation code</li> <li>● Allows more features than are available as standard.</li> <li>● The programmer may not have the expertise to develop that particular piece of code themselves.</li> <li>● It could speed up the development of their own projects by reducing the amount of developing and testing that developers would have to complete.</li> <li>● The libraries could be updated by an external body (to more efficient code for instance) thereby increasing the efficiency of the developers' own code.</li> </ul> <p><b>The disadvantages of using external code sources:</b></p> <ul style="list-style-type: none"> <li>● Programmer relies on another company to update their code</li> <li>● May reduce the security of the site.</li> <li>● External code may not be well documented.</li> <li>● Limited by what the external source of code offers.</li> <li>● Relying on the correctness of someone else's program takes control away from the developer.</li> <li>● You may need to become familiar with how to use the external code source (the API).</li> <li>● The external code source may cost money to use.</li> <li>● There is a risk of shoe-horning the project if the external code sources do not do exactly what the developer would like them to.</li> </ul>

<ul style="list-style-type: none"> <li>● <b>Database Concepts</b></li> </ul>	<p><b>Relational databases</b> consist of more than one table that are linked using key fields. A relational database has three main components; a store, a method of creating and retrieving data, and a method of ensuring that the data is logically consistent.</p> <p>Possible situations when a text file would be a better choice than a relational database:</p> <ul style="list-style-type: none"> <li>● When the data can be edited by users</li> <li>● When the data to be stored does not need the overhead of a relational database.</li> <li>● Could save memory space if data is not large.</li> </ul> <p><b><u>The tables of a relational database have the following features:</u></b></p> <ul style="list-style-type: none"> <li>● <b>Field:</b> It is the smallest piece of information that can be retrieved from a database and it is at the intersection of a row and a column in a database table.</li> <li>● <b>Foreign Key:</b> The foreign key is a column identified to establish a connection between the data in two tables. In the context of relational databases, a foreign key is a field (or collection of fields) in one table that uniquely identifies a row of another table</li> <li>● <b>Index:</b> used to speed up searches</li> <li>● <b>Primary Key:</b> uniquely identifies a row of data</li> <li>● <b>Record:</b> a row of data within a table</li> <li>● <b>Relationship:</b> links between tables</li> <li>● <b>Column:</b> A column within a table contains all the information of a single type.</li> <li>● <b>Search Criteria:</b> In databases you use search criteria to find records, using logical operators if there is more than one part to the criteria.</li> </ul>
<ul style="list-style-type: none"> <li>● <b>Query Methods (SQL)</b></li> </ul>	<p><b>Query</b> - this is a question posed to the database. This is done using SQL. <b>SQL</b> (Structured Query Language) is a programming language designed to manage data with a relational database.</p> <p>Fetching Data using SQL</p> <ul style="list-style-type: none"> <li>● <b>The SELECT, FROM clause:</b> <ul style="list-style-type: none"> <li>○ <b>SELECT * FROM table1</b> <ul style="list-style-type: none"> <li>● This retrieves all of the information from 'table1'</li> </ul> </li> <li>○ <b>SELECT Name, ID, D.O.B FROM table1</b> <ul style="list-style-type: none"> <li>● This retrieves data from columns 'Name', 'ID' and 'D.O.B' from 'table1'</li> <li>● The column will be shown in this order</li> </ul> </li> </ul> </li> <li>● <b>The WHERE clause:</b> <ul style="list-style-type: none"> <li>○ <b>SELECT Name, ID, D.O.B FROM table1 WHERE ID = 10</b> <ul style="list-style-type: none"> <li>● This does the same as before but only selects the data whose ID is not equal to 10 (= means equal to)</li> </ul> </li> </ul> </li> </ul>

	<ul style="list-style-type: none"> <li>● <b>ordering results from an SQL SELECT query:</b> <ul style="list-style-type: none"> <li>○ <b>ORDER BY column-name ASC</b> <ul style="list-style-type: none"> <li>● The ASC means the results are ordered alphabetically.</li> </ul> </li> </ul> </li>   <li>● <b>Adding Data using SQL</b> <ul style="list-style-type: none"> <li>○ <b>The INSERT INTO statement:</b> <ul style="list-style-type: none"> <li>▪ <b>INSERT INTO table (Name, ID, D.O.B) VALUES (Jim, NULL, NULL)</b> <ul style="list-style-type: none"> <li>● This insert the value 'Jim' into the 'Name' column in 'table1' but nothing is inserted for the column 'ID' and 'D.O.B'</li> <li>● NULL means there is no data in that field</li> </ul> </li> </ul> </li> </ul> </li>   <li>● <b>Editing Data using SQL</b> <ul style="list-style-type: none"> <li>○ <b>The UPDATE statement:</b> <ul style="list-style-type: none"> <li>● <b>UPDATE</b> tablename <b>SET</b> column1 = value1 , column2 = value2 , ... <b>WHERE</b> some_column = some_value</li> <li>● This edits the data in 'tablename' by setting 'column1' and the 'column2' field to a new value where 'some_column' is 'some_value'.</li> </ul> </li> </ul> </li> </ul>
<ul style="list-style-type: none"> <li>● <b>Connecting to databases from web applications and web based apps.</b></li> </ul>	<p>By storing data in web-based databases, new web technologies have led to 'distributed applications'.</p> <p>Distributed application programs have many parts that are stored on different virtual machines. The different virtual machines can be on the same or even different systems, allowing smartphones to access the same resources as desktop computers.</p> <p><b>Web-based software</b> is usually paid for on a subscription or usage basis, allowing companies to pay only what they need and to grow into the system without large upfront costs.</p> <p><b>Client-server</b> software is usually paid for upfront and the initial cost of the system can be high.</p> <p><b>Client server applications have advantages over web-based applications.</b></p> <ul style="list-style-type: none"> <li>● You are in control of upgrades.</li> <li>● You have to manage your own systems and pay for technical support.</li> <li>● Applications tend to run faster when they are local to the user's computer.</li> </ul>
<ul style="list-style-type: none"> <li>● <b>The use of computer technology in society</b></li> </ul>	<p>Our lives are fully integrated into a computer science world of mobile devices, tablet computers, desktops and laptop computers.</p> <p>Every day social media is changing the way we communicate and interact. It even plays an important motivational role in getting people involved in political and social debates.</p> <p>The advances in technology have also revolutionised medical treatments, and microprocessors can now even be used to control artificial limbs.</p> <p>Most devices are becoming web enabled, which means that they are connected to each other by the internet.</p>

Even software applications are becoming cloud based. The growth has led to many benefits and drawbacks including ethical issues and digital divide.

Ethical issues of advances in technology:

- Stealing software
- Plagiarising
- Making illegal or unethical use of ICT facilities
- Damaging, destroying, stealing and illegally using ICT facilities and files that belong to others.
- Cyberbullying
- Hacking
- Piracy

The term **digital divide** is the **gap** between people who can access technology and people who cannot. This could be as simple as a person being able to buy things cheaper online than someone without the technology but also refer to countries that have no easy access to the internet and all it offers in terms of education and knowledge.



**SP RESOURCES**

