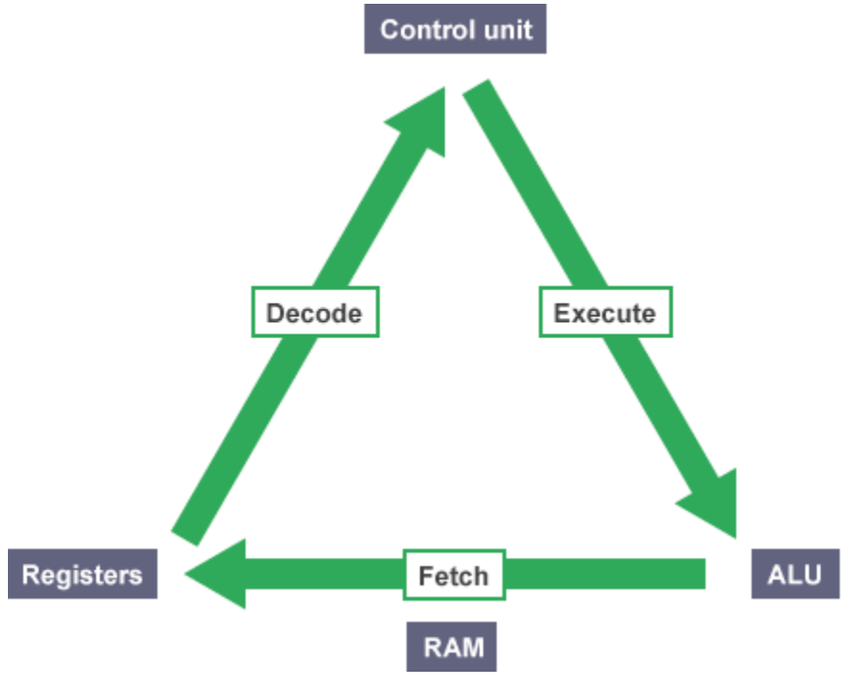
Knowledge Organiser - 1.1 Systems Architecture

Cs

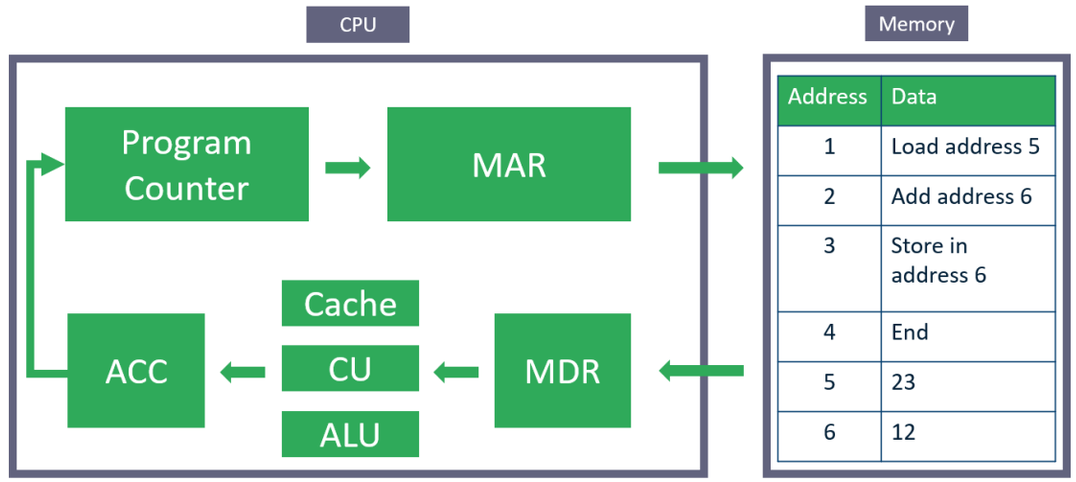
Key vocabulary

CPU	**Central Processing Unit.** Fetches – Decodes – Executes instructions.
CU	**Control Unit.** Manages the components of the CPU.
ALU	**Arithmetic and Logic Unit.** Carries out basic mathematics and comparisons.
Cache	Super fast memory located next to the CPU. Stores commonly used data & instructions.
Registers	Small memory containers inside the CPU.
MAR	**Memory Address Register**. Stores the address of the next instruction.
MDR	**Memory Data Register.** Stores the data fetched from the main memory (RAM).
Program Counter	A register that records the current instruction being executed.
Accumulator	A register used by the (ALU) to store the results of calculations.
Clock Speed	The speed of a computer's CPU, measured in hertz. This indicates the number of fetch-decode-execute cycles that can run per second.
Cores	A processing unit within a CPU. CPUs can have multiple cores.
Embedded System	A special purpose computer built into another device.

Fetch – Execute Cycle



Von Neumann Architecture



Knowledge Organiser - 1.2 Memory & Storage

Cs

Units

Unit	Abbreviation	Value
Petabyte	PB	1000^5 bytes
Terabyte	TB	1000^4 bytes
Gigabyte	GB	1000^3 bytes
Megabyte	MB	1000^2 bytes
Kilobyte	KB	1000 bytes
Byte	B	8 bits
Nibble	N	4 bits
Bit	b	0 or 1

Capacity Calculations

Moving between units	* By 1000 to move down / By 1000 to move up
Sound file size	Sample rate * duration * bit depth
Image file size	Colour depth * image height * image width
Text file size	Bits per character * number of characters

Key vocabulary

Primary Storage	Volatile memory used to store currently used data and instructions.
RAM	**Random access memory.** This is volatile memory that is constantly being written to and read from. It does not retain its contents without a constant supply of power. When a computer is turned off, everything stored in its RAM is lost.
ROM	**Read only memory**. This is non-volatile memory or storage containing data that cannot be changed.
Virtual Memory	A section of a secondary storage which is temporarily used as RAM.
Secondary Storage	Non-volatile memory used for long-term storage of programs and data.
Optical Storage	Storing and reading data from a disc using a laser. Examples include CD, DVD, Blu-ray.
Magnetic Storage	Storing and reading data from a hard drive disc using magnetism.
Solid State Storage	Storing and reading data using electricity
Capacity	The maximum amount of data that a device can contain.
Compression	A method of reducing file sizes, particularly in digital media such as photos, audio and video.
Lossy Compression	A form of compression that reduces digital file sizes by removing data.
Lossless Compression	A form of compression that encodes digital files without losing detail. Files can also be restored to their uncompressed quality.

Knowledge Organiser - 1.2 Memory & storage

Cs

Key vocabulary

Denary/Decimal	The number system most commonly used by people. It contains 10 unique digits 0 to 9. Also known as decimal or base 10.
Binary	A number system that contains two symbols, 0 and 1. Also known as base 2.
Hexadecimal	A number system using 16 symbols from 0-9 and A-F, also known as base 16 and hex.
Binary Shift	Multiply a binary number by shifting digits to left. Divide by shifting to the right. Fill gaps with zeros.
Character set	A table of data that links a character to a number. This allows the computer system to convert text into binary. Examples are ASCII and Unicode.
Pixel	Picture element - a single dot of colour in a digital bitmap image or on a computer screen.
Metadata	Data about data, eg photo image files have data about where the photo was taken and which camera took the picture.
Colour Depth	The amount of bits available for colours in an image.
Resolution	The fineness of detail that can be seen in an image - the higher the resolution of an image, the more detail it holds. In computing terms, resolution is measured in dots per inch (dpi).
Sample rate	How many samples of data are taken per second. This is normally measured in hertz, eg an audio file usually uses samples of 44.1 kHz (44,100 audio samples per second).
Bit depth	The number of bits available to store an audio sample.
Duration	Length of a file in terms of time. (minutes and seconds)

Binary Conversion

Find the largest value that fits into your number then subtract. Repeat.
The number 42 in binary:

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

Binary Addition

Start at the right hand side of any addition and follow the rules.
Here is 7+6 in binary. Note the carries go above the column to the left.

+	=
0+0	0
0+1	1
1+1	0 carry 1
1+1+1	1 carry 1

1	**1**		
0	1	1	1
0	1	1	0
1	1	0	1

Hexadecimal conversion

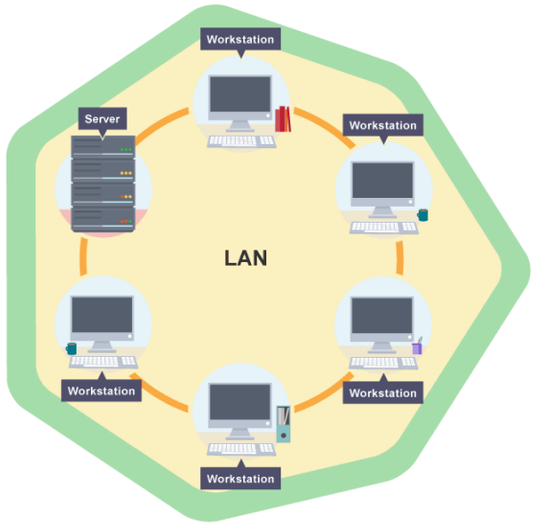
The hexadecimal number system is 0-9 then A-F (A represents 10)
Carry out a binary conversion then split the number into 2 nibbles. Then convert the two separate values into hexadecimal.
The denary number 42 in Hexadecimal is 2A

128	64	32	16	8	4	2	1
0	0	1	0	1	0	1	0
8	**4**	**2**	**1**	**8**	**4**	**2**	**1**
0	0	1	0	1	0	1	0
In hex 2 is 2				In hex 10 is A			

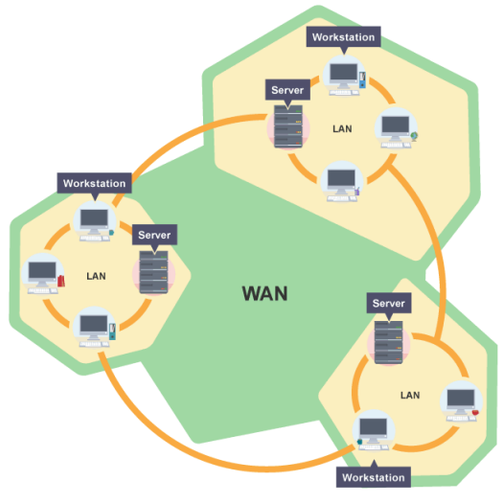
Knowledge Organiser - 1.3 Networks

Types of network

LAN

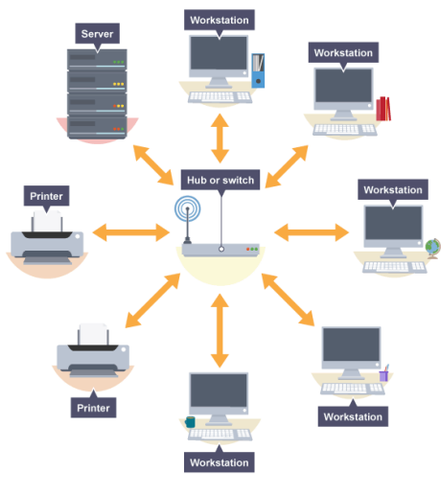


WAN

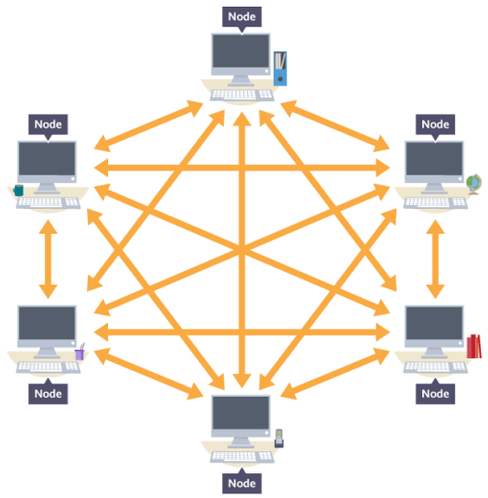


Topologies

Star



Mesh



Key vocabulary

Network	A group of interconnected computers/devices.
LAN	Local area network. A network of computers that covers a small area, eg a school or college.
WAN	Wide area network. A network that spans across a building, buildings or even countries, eg the internet.
Client-server	A relationship in which data or web application is hosted on a server and accessed by client computers.
Peer to peer	A relationship where all computers on the network share responsibility and there is no one central server.
WAP	A device that connects computers to a network using Wi-Fi.
Switch	A device for connecting computers and other network capable devices together to form a network.
NIC	**Network Interface Controller** -A circuit board that is installed in a computer so it can be connected to a network.
Transmission media	How data is carried from point A to point B physically, either by cable or wirelessly.
Ethernet	A set of protocols used in a wired local area network that describes how data is transmitted within it.
Wi-Fi	A method of connecting to the internet wirelessly using radio waves.
Bluetooth	Wireless technology used for transmitting data over short distances.
DNS	**Domain name server** - an internet service that translates IP addresses into website domain names. All websites have equivalent IP addresses.
Host	A server that stores files for other computers to access.
Cloud	A term often used to describe a location on the internet from which software applications are run and where data is stored.

Knowledge Organiser - 1.3 Networks

Cs

Key vocabulary

Encryption	Files that are encrypted have been altered using a secret code and are unreadable to unauthorised parties.
IP address	A unique address for each computer device on a network.
MAC address	Media access control - each unique piece of hardware on a network has a MAC address.
Standard	An agreed way of doing things.
Protocol	A set of rules for how messages are turned into data packets and sent across networks.

Layers

Layering means to break up the sending of messages into separate components and activities. Each component handles a different part of the communication. This can be referred to as the Transmission Control Protocol/Internet Protocol (TCP/IP) model.

Layering allows **standards** to be developed, but also to be adapted to new hardware and software over time. For example, different software packages (applications) may use the same transport, network and link layers but have their own application layer. The way the program encodes the message changes - the rest of communication method remains the same.

Common protocols

TCP/IP	**Transmission Control Protocol/Internet Protocol** - enables communication over the internet.
HTTP	**Hypertext Transfer Protocol** - governs communication between a webserver and a client.
HTTPS	HTTPS **(secure)** includes secure encryption to allow transactions to be made over the internet.
FTP	**File Transfer Protocol** - governs the transmission of files across a network and the internet.
POP	**Post Office Protocol** – governs the transmission of emails to devices. Once downloaded to the device is deleted from the server.
IMAP	**Internet Message Access Protocol** – governs the transmission of emails. Stored on server and accessed by devices.
SMTP	**Simple Mail Transfer Protocol** - governs the sending of email over a network to a mail server.
Layering	In networking, the concept of breaking up communication into separate components or activities.

Encryption

A simple method of encryption requires the use of a technique known as the Caesar cipher. The cipher works by giving a number value to a key. Each plaintext letter is replaced by a new letter, the one found at the original letter's position in the alphabet plus the value of the key. The example uses a key value of 3.

Plaintext	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
Ciphertext	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c

Knowledge Organiser - 1.4 Network Security

Cs

Protecting networks

Form of attack	Prevention
Malware	Anti-Malware software.
Phishing	Training of user to detect scams as well as the filtering of emails.
Brute-force attacks	Use of different strong passwords. A limit on the number of incorrect attempts.
Denial of service attacks	Block IP addresses which send too many requests. Increase capacity.
Data interception and theft	Encryption of data.
SQL injection	Ensuring that all data input is sanitized. (Forcing data to be in the format you want it such as a date, text or integer.)

Key vocabulary

Malware	Software that is designed to cause harm or damage to a computer. This includes viruses that might damage files, adware that causes pop-ups, and spyware that collects and shares login details.
Social Engineering	Tricking people into giving sensitive data such as PINs or passwords.
Phishing	An attempt to gain personal information about someone by way of deception, eg sending an email pretending to be from their bank asking them for their bank details.
Brute-force attack	Attempting every combination of a password or encryption key until it is correct.
Denial of service attack	An attack designed to render online services inaccessible. One type of this attack involves many computers simultaneously flooding a target with network traffic.
Data interception	Where data is intercepted during transmission. This is done using software called a packet sniffer, which examines data packets as they are sent around a network.
SQL Injection	Where SQL code is entered as a data input. Many databases use SQL code to interrogate the data and maintain the structure. SQL code can be inputted as data, which can cause errors or unintended operations.
Penetration testing	Systems are tested for vulnerabilities to reveal any weaknesses in the system which can be fixed.
Anti-malware	A type of computer program which detects, prevents and removes malware on a system.
Firewall	An application that prevents unauthorised connections to and from the Internet.
User-access level	These are the permissions given to a user to access facilities on a computer.
Encryption	Files that are encrypted have been altered using a secret code and are unreadable to unauthorised parties.

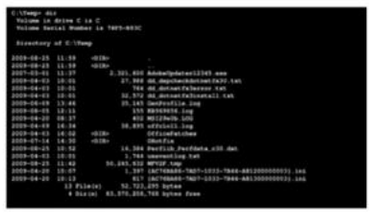
Knowledge Organiser - 1.5 Systems Software

Cs

Key vocabulary

Operating System	The software that manages the hardware and software resources in a computer system.
User interface	The means by which a user interacts with a computer or device.
Memory Management	Managing which data is stored in which location in the main memory.
Multitasking	In computing, running more than one program simultaneously.
Peripheral	A piece of hardware that connects to a computer, eg a mouse, keyboard, printer or scanner.
Driver	Software that controls and communicates with peripherals
User management	Organising how user interfaces and data are represented to different users.
File management	Organising how data is stored on secondary storage.
Utility software	A program which performs important maintenance tasks to improve the performance of a computer system.
Encryption software	Files that are encrypted have been altered using a secret code and are unreadable to unauthorised parties.
Defragmentation	The process of reordering files stored on a hard disk so that their segments run contiguously.
Data Compression	A method of reducing file sizes, particularly in digital media such as photos, audio and video.

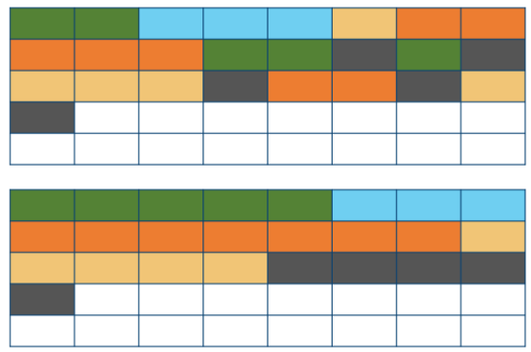
User interfaces

	A **Graphical User Interface (GUI)** is the most common interface. It uses Windows, Icons, Menus and a Pointer to display data and allow the user to navigate. It requires processor power to run. It is easy to use and requires little training.
	A **Command Line interface (CLI)** uses text to display data only and accepts only text commands from the user. It requires minimal processor power to run. It requires the user to learn commands to use.
	A menu driven interface uses a series of menus in a tree to display data and allow the user to navigate. It requires no training to used. Examples include ticket machines.
	A Natural Language interface uses linguistic commands that are spoken in order to allow the user to navigate and represent data. Processing power is required to understand the user request and respond. No training is required in order to use.

Defragmentation

Over time files stored can become fragmented.

Defragmentation removed any gaps and reorders data so that files are contiguous and all free space is at the end.



Knowledge Organiser - 1.6 Ethics

Cs

Impacts of digital technology

Ethical issues

Ethics are moral principles, or rules, which govern a person's attitudes and behaviour. Ethics apply to the use of computers as much as they do to other things in life. Ethical issues in computing include:

- Ensuring public safety
- Security of data

Cultural issues

The introduction of computers has changed society, sometimes for the better, sometimes for the worse. 'Cultural issues' is the term used for computer matters that have an effect on the nature and culture of society. Some of these issues include:

- The digital divide
- The changing nature of employment

Environmental issues

- Resources are needed to in order for computers to be produced, distributed and used. Metals and plastics are used to manufacture components, while energy is expended in distributing equipment and in using it.
- Many computers, such as web servers, domain name servers and data centres, need to be left running continuously. This requires lots of energy to maintain.
- Many computer components are either hard to recycle or contain toxic materials, such as lead.

Privacy issues

As more and more services become digitised users are worried about the amount of data organisations and governments gather. Eg Google Maps stores all data locations permanently unless opted out. Google know exactly where their users are, have been and for how long. This data is used for helpful purposes but it could also potentially be abused.

Legislation relevant to Computer Science

The Data Protection Act 2018

This law protects your data when used by companies and organisations. Personal data must be:

Fairly & lawfully processed	Obtained for legitimate purposes
Adequate, relevant & not excessive	Accurate & up to date
Not be kept longer than necessary	Handled securely.

Computer Misuse Act 1990

There are three separate parts to the Act:

1. It is illegal to access a computer unless you have permission to do so.
2. It is illegal to access data on a computer when that material will be used to commit further illegal activity, such as fraud or blackmail.
3. It is illegal to make changes to any data stored on a computer when the user does not have permission to do so.

Copyright Designs and Patents Act 1988

The **Copyright, Designs and Patents Act 1988** exists to protect peoples' creations. When a person creates something, they own it. What they create might include:

- a picture, drawing or photograph
- a video, television programme or film
- text, such as a book, article or report
- a game

Software licences

Open Source	Proprietary
Free of copyright	Copyright protected
Generally free	May have a fee or subscription licence
Source code is public	Source code is not public
Source code can be modified	Updates from manufacturer only
Possible online support from peers.	Can be expensive
Support may not be available	Cannot be modified
No guarantee of quality	Licence can limit number of installs

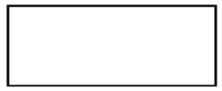
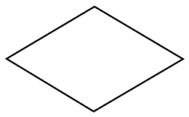
Knowledge Organiser - 2.1 Algorithms

Cs

Key vocabulary

Algorithm	A sequence of logical instructions for carrying out a task. In computing, algorithms are needed to design computer programs.
Computational Thinking	A problem-solving method using computer science techniques, where possible solutions are developed and presented in a way that can be understood by humans and computers.
Abstraction	The process of extracting or withdrawing something.
Decomposition	Breaking down a complex problem or system into smaller parts that are more manageable and easier to understand.
Pseudocode	Also written as pseudo-code. A method of writing up a set of instructions for a computer program using plain English. This is a good way of planning a program before coding.
Trace table	Trace tables are used to allow programmers to trace the value of variables as each line of code is executed. The values of the variables are displayed in a table and assist the programmer in identifying any potential errors.

Flowcharts

	Line		Input/output
	Process		Decision
	Sub Program		Terminal

Searching Algorithms

Linear Search

A linear search is the simplest method of searching a **data** set.
Starting at the beginning of the data set, each item of data is examined until a match is made. Once the item is found, the search ends.

1. A way to describe a linear search would be:
2. Find out the length of the data set.
3. Set counter to 0.
4. Examine value held in the list at the counter position.
5. Check to see if the value at that position matches the value searched for.
6. If it matches, the value is found. End the search.
7. If not, increment the counter by 1 and go back to step 3 until there are no more items to search.

A linear search, although simple, can be quite inefficient. Suppose the data set contained 100 items of data, and the item searched for happens to be the last item in the set? All of the previous 99 items would have to be searched through first.
However, linear searches have the advantage that they will work on any data set, whether it is ordered or unordered.

Binary Search

A binary search is an efficient method of searching an ordered list.
A binary search works like this:

1. Start by setting the counter to the middle position in the list.
2. If the value held there is a match, the search ends.
3. If the value at the midpoint is less than the value to be found, the list is divided in half. The lower half of the list is ignored and the search keeps to the upper half of the list.
4. Otherwise, if the value at the midpoint is greater than the value to be found, the upper half of the list is ignored and the search keeps to the lower half of the list.
5. The search moves to the midpoint of the remaining items. Steps 2 through 4 continue until a match is made or there are no more items to be found.

Sorting Algorithms

Bubble Sort

Bubble sorts work like this:
Start at the beginning of the list. Compare the first value in the list with the next one up. If the first value is bigger, swap the positions of the two values. Move to the second value in the list. Again, compare this value with the next and swap if the value is bigger. Keep going until the there are no more items to compare. Go back to the start of the list.

Each run through the list, from start to finish, is known as a pass. The bubble sort continues until a pass is made where no values have been swapped. At this point, the list is sorted.

Consider this unsorted list:

Position in list	0	1	2	3	4
Data value	9	4	2	6	5

The value at position 0 is 9, and the value at position 1 is 4. 9 is bigger than 4, so the two items would be swapped. The list would now be:

Position in list	0	1	2	3	4
Data value	4	9	2	6	5

We move up to the next position, position 1. The value at position 1 is 9, and the value at position 2 is 2. 9 is bigger than 2, so the two items would be swapped. The list would now be:

Position in list	0	1	2	3	4
Data value	4	2	9	6	5

We move up to the next position, position 2. The value at position 2 is 9, and the value at position 3 is 6. 9 is bigger than 6, so the two items would be swapped. The list would now be:

Position in list	0	1	2	3	4
Data value	4	2	6	9	5

We move up to the next position, position 3. The value at position 3 is 9, and the value at position 4 is 5. 9 is bigger than 5, so the two items would be swapped. The list would now be:

Position in list	0	1	2	3	4
Data value	4	2	6	5	9

The first pass is now complete. However, this list may still be unsorted, so another pass takes place.

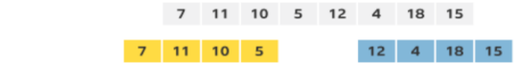
Merge Sort

A merge sort is a more complex sort, but also a highly efficient one.

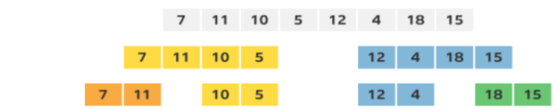
A merge sort uses a technique called divide and conquer. The list is repeatedly divided into two until all the elements are separated individually. Pairs of elements are then compared, placed into order and combined. The process is then repeated until the list is recompiled as a whole.

Consider this unsorted list:

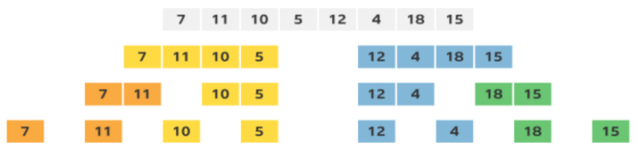
The list is split into half:



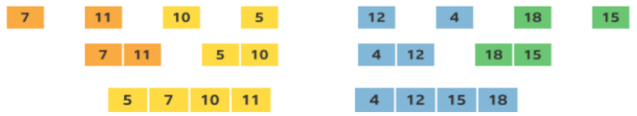
The process repeats:



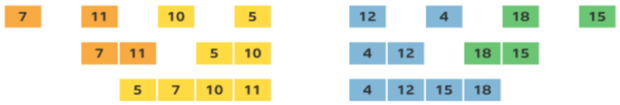
Until all elements are individually separated:



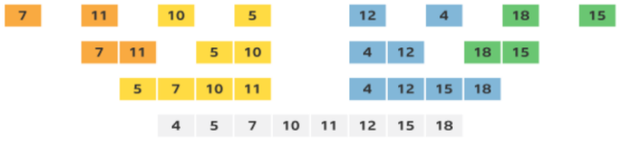
The **algorithm** looks at the individual elements and compares them as pairs. Each pair is sorted into order:



The process is repeated for the initial right hand division:



Eventually the list is recompiled



The list is now sorted into the correct order.

Insertion Sort

An insertion sort is less complex and efficient than a merge sort, but more efficient than a bubble sort. An insertion sort compares values in turn, starting with the second value in the list.

If this value is greater than the value to the left of it, no changes are made. Otherwise this value is repeatedly moved left until it meets a value that is less than it.

The sort process then starts again with the next value. This continues until the end of the list is reached.

Consider this unsorted list:

12 is the first value in the list. No other values are before it, so the sort moves on to the next value, 14. 14 is greater than the value to the left, 12, so the sort moves on again to the next value, 13. 13 is less than 14, so the two elements are swapped:

13 is greater than 12, so the sort moves onto the next value, 14.
14 is greater than 13, so again the sort moves on.

11 is less than 14, so the two values swap:

11 is less than 13, so the two values swap:

11 is less than 12, so the two values swap:

The process repeats:

Knowledge Organiser - 2.2 Programming Fundamentals

Cs

Key vocabulary		Python Syntax
Variable	A memory location within a computer program where values are stored.	**player_age** = (15)
Constant	A value that does not change when the program is running.	pi = (3.14)
Assignment	Setting the value of a variable in a computer program.	player_age **= (15)**
Input	Data received by a program.	player_name = **str(input("What is your name?"))**
Output	Data sent out of a program.	**print** (player_name)

Data types		Python Syntax
Integer	Whole number	player_age = (**15)**
Real	Decimal number. Called a float in Python.	player_height = (**150.8**)
Boolean	True or False	player_active – (**True**)
String	A sequence of characters which include letters, numbers & symbols.	player_name = (**"Zaphod"**)
Casting	Changing the data type of a variable.	int(4.2)

Comparison Operators

==	Equal to	**!=**	Not equal to
>	Greater than	**>=**	Greater than or equal to
<	Less than	**<=**	Less than or equal to

Arithmetic Operators

+	Addition	**-**	Subtraction
*****	Multiplication	**/**	Division
MOD	Modulus	**DIV**	Quotient
^	Exponentiation (to the power)		

Imports

Time – allows you to pause your program.	import time #---------- time.sleep(1)
Random – Gives the ability to generate a random integer.	import random #---------- secret_num = (random.randint(1,10))

Length of a string	Python syntax
The length of a string can usually be determined using the len statement. This gives the length as an integer.	word = (“Computer”) word_length = (len(word))

Character position	Python syntax
It is possible to determine which character features at a position within a string as each character is numbered. Computers start counting at 0 so the first character is always 0.	word = (“Computer”) print (word[2]) #would print the character “m” as c = 0 and o = 1.
Several characters can be determined using a range of characters.	word = (“Computer”) print (word[3:8]) #would print “puter”

Upper and lower case	Python syntax
It is possible to change all letters in a string to either lowercase or uppercase. This can be very useful, for example when checking possible inputs.	word = (“Computer”) word = word.upper() #would change the string to “COMPUTER”
	word = (“Computer”) word = word.lower() #would change the string to “computer”

Concatenation	Python syntax
To concatenate strings means to join them to form another string – adding two strings together.	word = (“Computer”) sentence = (word + “Science”) #would add the two strings together to form one string which is “Computer Science”

Knowledge Organiser - 2.2 Programming Fundamentals

Cs

Sequence	Python syntax
A set of instructions or lines of code that follow on one from another.	player_name = str(input("Enter your name") print ("Hello", player_name)

Selection	Python syntax

A decision within a computer program when the program decides to move on based on the results of an event.	player_colour = int(input("Press 1 for red, 2 for blue or 3 for green"))
	if player_colour == 1:
	print ("You have selected red")
	elif player_colour == 2:
	print ("You have selected blue")
	elif player_colour == 3:
	print ("You have selected green")
	else:
	print ("You did not press 1,2 or 3")

Iteration (while)	Python syntax
The repetition of a block of statements within a program when the number of repeats is not known.	player_turns = 5 while player_turns >0: print ("Player takes their turn") player_turns = player_turns – 1
	active = True while active == True: print ("Player is active")

Iteration (for)	Python syntax
The repetition of a block of statements within a program when the number of repeats is known.	for counter in range (0,10,1): print (counter)

Sub Programs		Python syntax
Procedure	A section of computer code that performs a specific task.	def greeting (): print (“Hello world”) #---------- greeting()
Function	A section of code that, when programming, can be called by another part of the program with the purpose of returning one single value.	def addition(a,b): c = (a+b) return (c) #---------- num1 = int(input(“Enter first number:”)) num2 = int(input(“Enter first number:”)) answer = (addition(num1,num2)) print (num1,”+”,num2,”=“,answer)

Arrays		Python syntax
1D Array	An array is a data structure that holds similar, related data. An array is like a collection of boxes, each of which is called an **element**. Each element has a position in the array, and can hold a value. The data in an array must all be of the same **data type**. 0: Pen; 1: Pencil; 2: Ruler; 3: Eraser	stationery = [“Pen” , “Pencil” , “Ruler” , “Eraser”] print (stationery) #would display whole array print (stationery[0]) #would output the first element – “Pen” print (stationery[0-2]) #would output the first 2 elements – “Pen,Pencil”
2D Array	A two-dimensional array can hold more than one set of data. This type of array is like a table, with data held in rows and columns. 0: Pen, Pencil, Ruler, Eraser 1: Red, Green, Blue, Yellow	stationery = [[“Pen”,“Pencil”,“Ruler”,“Eraser”] , [“Red,”Green”,”Blue”,”Yellow]] print (stationery) #would print the whole 2D array print (stationery[0]) #would print the first row only (the original stationery) print (stationery[1]) # would print the second row only (the coloured pens)

File handling		Python syntax

Open	A file must be open before a record can be read from or written to it. To open a file, it must be referred to by its **identifier** within the program. Files can be opened in read mode (r), write (w) or append (a)	file = open(“scores.txt” , “r”) #would import the contents of score into the variable file in read only mode
		file = open(“scores.txt” , “a”) #would import the contents of score into the variable file in append mode
		file = open(“scores.txt” , “w”) #would import the contents of score into the variable file in write mode

Read	Once a file has been opened, the records are read from it one line at a time. The data held in this record can be read into a **variable**, or, more commonly, an **array**	score = file.read() #reads the entire file
		score = file.readline() #reads a single line

Write	Data is written to a file one line at a time, using the writeLine statement	for x = 0 to 9 file.write(scores[x])
Closing	A file must be closed by the program for it to be saved.	file.close()

SQL

SQL

Structured query language (SQL)
Databases use their own type of programming language. This language is known as structured query language, or SQL.

SELECT , FROM , WHERE

Data can be retrieved from a table using these commands.

* Stands for wildcard which means all records.

Table name = Computer_Scientists

ID	Title	Forename	Surname	Nationality
001	Mrs	Ada	Lovelace	GB
002	Mr	Alan	Turing	GB
003	Miss	Grace	Hopper	USA
004	Mr	George	Boole	GB

SELECT * FROM Computer_Scientists WHERE Title = “Mr”
#would return records 002 and 004

SELECT * FROM Computer_Scientists WHERE Nationality = “USA”
#would return record 003

SELECT * FROM Computer_Scientists WHERE Nationality = “USA”
#would return record 003

SELECT * FROM Computer_Scientists WHERE ID < “003”
#would return record 001 and 002

Knowledge Organiser - 2.3 Producing Robust Programs

Cs

Key vocabulary

Input Sanitation	The process of checking entered data and removing dangerous inputs which could otherwise be used to cause damage to a program.
Authentication	Verifying the identity of a user.
Validation	Checking input data is sensible and in the right format.
Maintainability	The process of ensuring that a program is easy to understand, modify and update.
Naming conventions	Choose either CamelCase or under_score conventions for variables and stick to one.
Comments	Annotation using a # in Python. Should be used to explain what a section of code intends to do.
Iterative testing	Tests carried out while a program is being developed. Step by step.
Final/terminal testing	A test carried out when all parts of a program are complete.
Syntax Error	Error in a program resulting from code not following syntax rules governing how to write statements in a programming language.
Logic Error	Error in a program which does not cause a program to crash but causes unexpected results.
Test data	Data entered into a program to test if it is working.
Normal data	Data entered into a program that should produce a positive result
Boundary data	Data entered into a program at the edge of its acceptable range
Invalid data	Data entered into a program that should produce a negative result.
Erroneous data	Data that a program cannot process and should not accept.

Test tables

Tests are laid out in a testing table, which indicates:

- The test number
- A description of what the test intends to check
- The test data being used
- The type of test - normal, boundary or erroneous
- Expected outcome
- Actual outcome

Ideally, a programmer should run as many tests as is sensible. Many large programs, especially games, contain bugs simply because it may not be possible to test every possible input or action.

Example

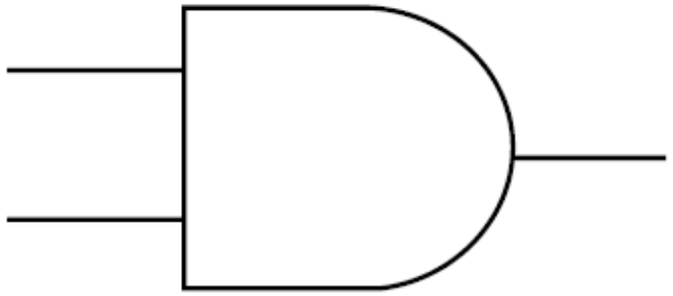
#	Description	Data	Test type	Expected	Actual
1	Number accepted?	5	Normal	Accept	Accept
2	Test low boundary	1	Boundary	Accept	Accept
3	Test upper boundary	10	Boundary	Accept	Accept
4	Test < boundary	-5	Erroneous	Reject	Reject
5	Test > boundary	20	Erroneous	Reject	Reject

Knowledge Organiser - 2.4 Boolean Logic

Cs

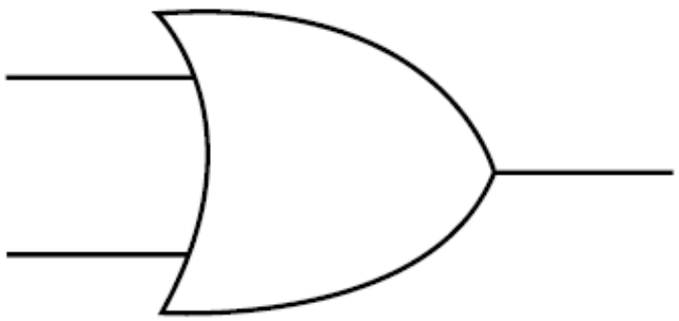
Logic Gates

AND (Conjunction)



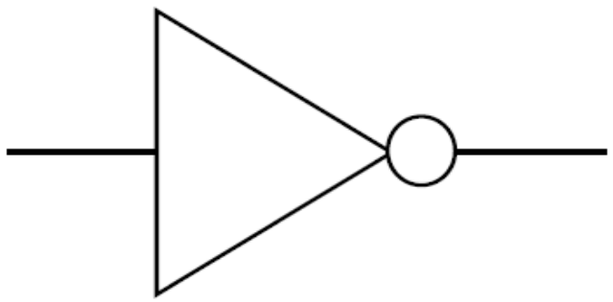
A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

OR (Disjunction)



A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

NOT (Negation)



A	B
0	1
1	0

Knowledge Organiser - 2.5 Programming Languages and IDE's

Cs

Key vocabulary

High level programming	Also known as high level language. This is a computer programming language used to write programs. High-level languages need to be translated into machine code through a compiler, interpreter or assembler.
Low level programming	Also known as low level language. This is a computer programming language which closely represents machine language. Low-level languages are more difficult to understand than high-level languages but they execute quicker.
Translator	Translators are needed to translate programs written in high level languages into the machine code that a computer understands. Tools exist to help programmers develop error-free code.
Assembler	A program that translates assembly languages into machine code.
Compiler	A program that translates high-level programming languages into machine code.
Interpreter	A program that translates high-level programming languages into machine code. Programs can either be interpreted or compiled.
IDE	A suite of tools that helps a programmer to write error-free, maintainable code.
Editor	The part of an IDE that allows the user to write and amend code.
Error Diagnostics / debug	The process of finding and correcting programming errors.
Run time environment	The part of an IDE where a program runs.

Example of an IDE