Knowledge organiser - Computing - Year 7 - Autumn 1

National Curriculum Strand:

- ☐ Computer Science
- ☐ Digital Literacy
- ■ Information Technology

Key Computing concepts

Productivity tools and office applications

Careers that might use this knowledge

Journalist

Researcher

Student

Progression path

I can explore and experiment with new programs to figure out their use based on my previous experience and by using available help options

I can use a range of advanced tools within the programs I use

I can choose the most appropriate tool for tasks and explain this choice

I can explain the main purpose of each Office application and make basic use of them

I can choose programs for drawing, writing and presenting.

I recognise that different programs have different purposes.

I know that computers have a range of programs on them.

Key Vocabulary

Microsoft Office	GSuite	Home Directory	Word Processor
Spreadsheet	DTP (Desktop Publishing)	Email	Inbox
Branching File Structure	Creative Commons	Network	Internet
Web browser	Application	Productivity Tool	HTML / WYSIWYG

Prior knowledge:

In Key Stage 2, pupils will have used a range of different programs. They can should understand how to save files in given locations and go back to files that they have previously made. They can carry out basic keyword searches for relevant information and make some judgements regarding reliability of information found.

Leading to:

Students should be able to use branching file structures independently and name files for reuse, recognising the importance of file names for purpose. They can choose the best tool for the job and compare the benefits and drawbacks of other programs. They can offer suggestions of different programs of the same type, e.g. word processors or presentation software. They will use advanced tools to search with more precision and show an awareness of usage rights and copyright issues.

Key Questions pupils should have the knowledge to answer

What is a word processor? A program for storing, manipulating, and formatting text entered from a keyboard and providing a printout. It's very useful for writing reports, essays, letters, and more. You should not use a word processor if a document is to include many images. Popular word processors include Microsoft Word, Google Docs, and Apple Pages.

What is a spreadsheet? An electronic document in which data is arranged in the rows and columns of a grid and can be manipulated and used in calculations. Spreadsheets are incredibly useful for tasks like budgeting, tracking expenses, managing inventories, and analysing data. By using formulas and functions, you can perform calculations, create charts and graphs to visualize data, and sort or filter information to make it easier to understand. Popular spreadsheet programs include Microsoft Excel, Google Sheets, and Apple Numbers. They help users to efficiently manage and interpret large amounts of data with ease.

What is desktop publishing? Desktop publishing (DTP) is the creation of documents using page layout software on a personal ("desktop") computer. The purpose of desktop publishing is to create high-quality documents that combine text and images, such as brochures, newsletters, books, and magazines. Desktop publishing software allows you to design and lay out pages in a professional manner, making it easier to control the appearance of the final product. This software typically includes tools for formatting text, arranging graphics, and incorporating various design elements, enabling users to produce polished, visually appealing publications. Popular desktop publishing programs include Adobe InDesign, Microsoft Publisher, and Canva. These tools help individuals and businesses produce print and digital publications efficiently and cost-effectively.

What is a home directory? A home directory is a file system directory on a multi-user operating system containing files for a given user of the system. Basically, it is a private (although administrators can access it) area for your files.

What must I consider when searching for information and images online? You need to should use reputable sources like educational institutions, verify information by checking multiple sources, and ensure the content is current. It's crucial to pay attention to the author's credentials. For images, trusted websites should be used, copyright laws respected, and authenticity verified. Safety and privacy are important; never share personal information and prefer secure websites (HTTPS). Critical thinking is key: consider biases and understand the search purpose to focus on reliable sources. When using images, it's important to respect copyright laws and look for materials under Creative Commons licenses, which allow for use with certain conditions, making it easier to share and use images legally and ethically.

What is a WYSIWYG editor (web)? A WYSIWYG editor for websites is a tool that allows you to create and design web pages without needing to write any code. "WYSIWYG" stands for "What You See Is What You Get." This means that what you see in the editor as you're creating your web page is exactly how it will look when published online. It's like using a word processor where you can add text, images, and other elements directly, and immediately see how they will appear on the actual website. This makes it much easier to build websites, especially if you don't know how to code. Examples of WYSIWYG editors include tools like Google Sites and Wix.

Do you want to know more?

https://www.youtube.com/watch?v=UFW-A-2-sM4

https://www.goskills.com/Microsoft-Office

Excel for beginners

Creating HTML website with the Caveman

https://www.youtube.com/watch?v=yCVy5Kw0l8s

Using Google Sites

https://learn.microsoft.com/en-us/office365/servicedescriptions/office-applications-service-description/office-applications

Knowledge organiser - Computing - Year 7 - Autumn 2

National Curriculum Strand:

- [x] Computer Science
- [ ] Digital Literacy
- [ ] Information Technology

Key Computing concepts

Computational Thinking

Basic programming concepts and constructs

Careers that might use this knowledge

Software developer

Financial Analyst / Stockbroker

Research scientist

Progression path

Assess their understanding of computational thinking concepts and areas for improvement. They also consider how they can apply their skills in other subjects and real-life situations.

Explore advanced programming concepts, such as functions, procedures, and data structures. Write more complex programs and understand how to solve more challenging problems.

Know basic programming concepts in Scratch. Able to write simple programs, understand syntax, and use variables, loops, and conditionals to control the flow of their programs.

Apply computational thinking concepts, with guidance, to real world problems, including programming.

Pupils learn to create step-by-step instructions (algorithms) to solve problems. They practice writing simple algorithms and understand the importance of clear and precise instructions.

Introduced to the basic concepts of computational thinking, including decomposition, pattern recognition, abstraction, and algorithms. They learn to identify problems and break them down into smaller, manageable parts.

Recognise that humans can solve problems using a range of techniques

Key Vocabulary

Algorithm	Abstraction	Decomposition	Pattern Recognition
Solution	Procedure / Subroutine	Trigger	For Loop
While Loop	Infinite Loop	Logic	Program
Iteration	Evaluate	Sequencing	Computational thinking

Prior knowledge:

Pupils learn the basics of computational thinking, including how to break down problems into smaller, manageable parts (decomposition) and recognize patterns to solve problems more efficiently. They are introduced to algorithms, understanding how to create step-by-step instructions for tasks, and begin to use simple programming languages to implement these algorithms. Additionally, pupils learn to debug their code, identifying and fixing errors to ensure their programs work correctly.

Leading to:

Pupils delve deeper into computational thinking by solving complex problems and developing sophisticated algorithms. They learn advanced programming concepts, including data structures, functions, and object-oriented programming, using Python to implement these concepts effectively. Additionally, pupils enhance their skills in debugging, testing, and evaluating their programs, preparing them for further studies and careers in computer science and related fields.

Key Questions pupils should have the knowledge to answer

What is computational thinking? Computational thinking is a way to solve problems by breaking them down into smaller parts, finding patterns, and creating step-by-step plans (algorithms) to solve them. It's a key skill in computer science but can be used in many other areas too. By thinking this way, you can approach problems methodically and come up with efficient solutions.

What are the core concepts of computational thinking? Decomposition: This means breaking a big problem into smaller, easier-to-handle parts. It's like taking a big puzzle and working on one piece at a time. **Pattern Recognition:** This is about finding similarities or patterns in problems. It's like noticing that every time you see a certain shape, it fits in a specific spot in your puzzle. **Abstraction:** This means focusing on the important details and ignoring the rest. It's like looking at a map and only paying attention to the roads you need to get to your destination. **Algorithm design:** These are step-by-step instructions to solve a problem. It's like following a recipe to bake a cake, where you do each step in order to get the final result.

What is the difference between a for loop and a while loop? A for loop repeats a block of code a specific number of times, making it useful when you know how many iterations you need. A while loop continues to repeat a block of code as long as a certain condition is true, which is helpful when the number of iterations isn't known in advance. An infinite loop will continue forever unless the code is interrupted.

What is a procedure / subroutine? A procedure is a set of instructions that perform a specific task. It does not return a value. Procedures are used to encapsulate code that performs a particular operation, making the code more modular and reusable. In some programming languages, procedures are called "subroutines." The term "subroutine" is often used in older programming languages like Fortran and BASIC.

Is a procedure the same as a variable? A procedure is a set of instructions that perform a specific task within a program. It is a block of code that can be called and executed whenever needed, but it does not store data like a variable does. A variable, on the other hand, is a storage location identified by a name (an identifier) that holds data values. Variables are used to store information that can be referenced and manipulated in a program.

How is computational thinking useful in computing? Computational thinking is super useful in computing because it helps you solve problems more easily. By breaking big problems into smaller parts, spotting patterns, and creating step-by-step plans (called algorithms), you can tackle different challenges in coding and beyond. These skills make it easier to write and fix code, and they prepare you for more advanced computer science topics in the future.

Do you want to know more?

https://www.bbc.co.uk/bitesize/topics/z7tp34j

https://www.youtube.com/watch?v=qbnTZCj0ugI&t=5s

https://www.safesearchkids.com/computational-thinking-for-children/

https://www.codemonkey.com/blog/the-advantages-of-computational-thinking-for-kids/

https://www.barefootcomputing.org/docs/default-source/at-home/quick--guide-to-computational-thinking.pdf

https://teachyourkidscode.com/what-is-computational-thinking/

Knowledge organiser - Computing - Year 7 - Spring 1

National Curriculum Strand:

- Computer Science
- Digital Literacy
- Information Technology

Key Computing concepts
Procedures
Subroutines
Triggers

Careers that might use this knowledge
App designer
Systems programmer
Toy developer

Progression path
I refine my work based on outcomes, efficiency, user feedback and suitability of purpose
I can create and use subroutines, procedures and functions within my programs
I use decomposition to create efficient solutions
I am aware of the importance of sequencing when programming instructions. I develop subroutines that interact through triggers
I can program series of instructions to be carried out
I can program instructions to be carried out
I can plan instructions to be carried out

Key Vocabulary

Program	Algorithm	Sequence	Iteration
Trigger	Procedure	Stepwise	Repeat
Loop	Variable	Decomposition	Efficient
Debug	Abstraction	Repetition	For/While

Prior knowledge:

Pupils will have experience designing, writing, and debugging programs to accomplish specific goals, including controlling or simulating physical systems. They will learn to use sequence, selection, and repetition in programs, work with variables, and use logical reasoning to explain how simple algorithms work and to detect and correct errors. Experienced triggers in PowerPoint. Used Scratch.

Leading to:

Text based programming of solutions using Python and physical computing with the Micro:Bits. Larger scale programs which require many sub-routines and independent outcomes. By the end of KS4 they will have a deep understanding of computer science principles. They will gain practical experience in designing, writing, and debugging programs to solve complex problems, using multiple programming languages and data structures. They will solve complex problems analytically and creatively

Key Questions pupils should have the knowledge to answer

What is the difference between the internet and the World Wide Web? The internet is a global network of interconnected computers and servers, enabling data communication between devices. The World Wide Web is a service that operates on the internet, providing access to websites and online content via protocols like HTTP/HTTPS. In simple terms, the internet is the infrastructure, and the web is a service built on top of it.

What is bandwidth, and why is it important? Bandwidth refers to the maximum amount of data that can be transmitted over a network connection in a given amount of time, usually measured in bits per second (e.g., Mbps or Gbps). It determines how quickly data can be sent or received. High bandwidth allows faster downloads, smoother streaming, and better performance when multiple devices use the network simultaneously. Low bandwidth can lead to slow speeds, buffering, or dropped connections.

What are the differences between wireless and wired networks? Wired Networks: Use physical cables (e.g., Ethernet) to connect devices. They are generally faster, more reliable, and less susceptible to interference. However, they limit mobility since devices must stay connected to the cable. **Wireless Networks:** Use radio waves (e.g., Wi-Fi) to transmit data. They allow greater flexibility and mobility but can be affected by interference, physical obstacles, or distance from the router. Wireless networks are also usually slower than wired networks due to shared bandwidth among devices.

How is data transmitted across a network using packets? Data is split into small units called packets for transmission across a network. Each packet contains a portion of the data along with metadata, such as the sender's and receiver's IP addresses and instructions for reassembly. The packets are sent individually across the network, potentially taking different paths, and are reassembled in the correct order at the destination.

What role do IP addresses play in data transmission? IP addresses are unique identifiers assigned to devices on a network. They act like a postal address, ensuring that data packets are sent to the correct destination and received from the correct source. For example, when you visit a website, your device's IP address tells the server where to send the data, and the website's IP address ensures you're accessing the right server.

How has the evolution of communication technologies, like from semaphores to the internet, impacted society? The evolution of communication technologies has drastically improved the speed, reliability, and accessibility of sharing information. Early methods like semaphores were limited in scope and speed, but the internet allows instantaneous global communication, enabling advances in education, commerce, healthcare, and entertainment. However, this evolution has also introduced challenges, such as cybersecurity threats and digital divides.

Do you want to know more?	
https://www.bbc.co.uk/bitesize/guides/zc6rcdm/revision/1	https://www.youtube.com/watch?v=3QhU9jd03a0
https://www.bbc.co.uk/bitesize/guides/z8nk87h/revision/1	https://www.youtube.com/watch?v=KjD3KANH-xc
https://www.bbc.co.uk/bitesize/articles/z78nydm	https://www.youtube.com/watch?v=mpFWIZ318eQ
https://aws.amazon.com/what-is/computer-networking/	https://www.youtube.com/watch?v=8sTy8466MoE

Knowledge organiser - Computing - Year 7 - Spring 2

National Curriculum Strand:

- [ ] Computer Science
- [ ] Digital Literacy
- [x] Information Technology

Key Computing concepts

Spreadsheet models

Databases

Sorting, searching and validating data

Careers that might use this knowledge

Accountant

Scientist

Market researcher

Progression path

Can explain the benefits and drawbacks of spreadsheets models in a business environment

Can use a wide range of efficiency tools. Can adapt and devise formulae to meet a need.

Can unpick models to identify how they work from the end point backwards

Can create models that meet the users needs

Can use formulae within cells to calculate answers

Can navigate cells in Excel and change formatting for purpose and clarity

Know key terms relating to spreadsheets: cell, row, column, format, formula

Key Vocabulary

Model	Cell reference	Look Up Table	Validation
Conditional	Criteria	IF	Formulae
Function	Efficiency tools	Reverse Engineer	Data vs. information
Data type	Field	Record	Spreadsheet

Prior knowledge:

In Key Stage 2, pupils have used simple formulae in spreadsheets and constructed models for home purposes (parties, saving etc.) Pupils can format cells, sort and filter data. They know what a model is and have worked with simple models to explore different scenarios.

Leading to:

Pupils should construct models that are fit for purpose using efficiency tools and advanced features. They will construct formulae with conditions or based on criteria (IF).

At KS 4, pupils are expected to understand spreadsheets. They should know how to store and manipulate data, use common functions like SUM and AVERAGE, utilise cell references, and create charts for data analysis. They should also solve problems and answer “what if…?” questions using spreadsheets.

WJEC Vocational Award in ICT focuses on technical skills, practical application, and real-world scenarios. The course combines theoretical and practical assessments to ensure students can effectively apply their knowledge in various ICT contexts.

Key Questions pupils should have the knowledge to answer

What is a spreadsheet? A spreadsheet is a digital tool composed of rows and columns used to organize information, perform calculations, and create charts for data visualization. It is highly effective for tasks such as budgeting, tracking progress, and analysing data.

Why do we use a formula in a spreadsheet? A formula in a spreadsheet is like a magic recipe that helps us do calculations. It tells the computer what to add, subtract, multiply, or divide. We use formulas to quickly find answers to math problems using the numbers in our spreadsheet.

How do spreadsheet models help us explore different scenarios? They enable you to model different scenarios and outcomes in several ways:

- What-If Analysis: Spreadsheets allow you to change variables and see how those changes affect the outcome. For example, you can adjust sales figures to predict revenue under different conditions.
- Scenario Analysis: You can create multiple scenarios within a spreadsheet to compare different sets of assumptions side by side. For example, you can model best-case, worst-case, and most-likely scenarios for a project.
- Data Visualization: Spreadsheets can create charts and graphs that help you visually analyze and compare different scenarios, making it easier to identify trends and patterns.
- Automated Calculations: By using functions and formulas, spreadsheets can automatically calculate results based on different inputs, saving time and reducing errors.

What advanced tools does Excel/Google Sheets have? Conditional formatting to influence cells based on conditions. Lookup tables where data is input from cells based on the criteria you 'look up'. Data validation to limit incorrect answers. Protection and locking to secure your spreadsheet. Drop down menus to ensure correct data is entered.

Why use a spreadsheet model? It allows storage and organisation of information so that you can access and understand it. Many features have been automated in a model so you can add new data for it to be calculated in the same way, thus saving time. An existing model can allow a novice to enter data and have it calculate and analyse for them.

What useful functions does a spreadsheet have? Spreadsheets have many functions to help you organize and analyse data. These include: **SUM -** Adds up a range of numbers. For example, =SUM(A1:A10) adds all numbers from cell A1 to A10; **AVERAGE—** Calculates the average of a group of numbers. For example, =AVERAGE(B1:B10) finds the average of numbers in cells B1 to B10. **MIN/MAX—**Finds the smallest or largest number in a range. For example, =MIN(C1:C10) finds the smallest number in cells C1 to C10. **IF -** Checks if a condition is true or false and gives different results. For example, =IF(D1>50, "Pass", "Fail") checks if the number in cell D1 is greater than 50.
What is data validation? Data validation is a feature in Excel used to control what a user can enter into a cell. For example, you could use data validation to make sure a value is a number between 1 and 6, make sure a date occurs in the next 30 days, or make sure a text entry is less than 25 characters. It stops users being able to enter invalid data.

Do you want to know more?

https://www.bbc.co.uk/bitesize/guides/zdydmp3/revision/1

https://www.youtube.com/watch?v=wbJcJCkBcMg

https://www.lynda.com/Excel-tutorials/Excel-2016-Advanced-Formulas-Functions/431188-2.html

https://www.excel-easy.com/

https://www.youtube.com/watch?v=kSQmPK-tWnw

https://alison.com/course/microsoft-excel-2019-beginners

https://support.microsoft.com/en-us/office/excel-video-training-9bc05390-e94c-46af-a5b3-d7c22f6990bb

Knowledge organiser - Computing - Year 7 - Summer 1

National Curriculum Strand:

- Computer Science
- Digital Literacy
- Information Technology

Key Computing concepts

- Computer networks
- Data transmission and packets
- Search engines

Careers that might use this knowledge

- Telecommunications engineer
- Web developer
- Network manager

Progression path

- Innovatively connects historical communication methods (e.g., semaphores) to advancing concepts like edge computing or AI-driven networks. Can critically evaluate potential future developments in communication technologies. Communicates understanding clearly, using precise terminology and real-world examples.
- Explores deeper implications of networking advancements (e.g., societal impacts of the internet).
  Analyses limitations and benefits of past and present technologies.
- Demonstrates detailed understanding with examples (e.g., describing how IP addresses enable communication). Applies understanding to explain modern technologies like 5G or IoT.
- Understand that data can be split into packets for transmission across networks. Explains terms accurately. Understands the role of networks and transfer protocols like TCP/IP.
- Shows a basic understanding of how networks work and terms like "data packet."
  Able to describe some differences between wired and wireless networks with guidance.
- Understand that hardware is needed to create computer networks
- Recognise that computers can share information with other computers

Key Vocabulary

Header	HTTP(S)	Packet	Protocol
Bandwidth	Network	IP Address	Router
Server	Ethernet	Web browser	Fibre optic
3g/4g/5g	Wifi	IoT	Search Engine

Prior knowledge:

KS2 students will have an awareness of historical and modern communication methods (e.g., postal systems vs. emails) and basic recognition of what a network is. They will understand that that data can be created, stored, and shared electronically (e.g., sending messages or using cloud platforms for projects) and know how files and folders can be used to save and access work.

Leading to:

This knowledge acts as a stepping stone for more complex topics in KS3 such as advanced networking concepts or the basics of cybersecurity. Establishing a basic understanding of concepts like packet-switching and the role of protocols is critical for tackling networking topics in the GCSE curriculum, as well as supporting students with real life technology and use of networks and devices at home.

Key Questions pupils should have the knowledge to answer

What is a procedure? A procedure is a set of coded instructions that tell a computer how to run a program or calculation. Many different types of programming languages can be used to build a procedure. Depending on the programming language, a procedure may also be called a subroutine, subprogram or function.

What is a trigger? A trigger controls an event. This can be timed automatically, reliant on an input or sensor. Trigger events can be broadcast in Scratch so that a 'message' is sent out once a certain part of the program is reached to initiate the next script or subroutine.

What is the difference between a procedure and a subroutine? A procedure is a set of instructions that performs a specific task and can be called by other parts of a program, while a subroutine is a more general term for any callable sequence of instructions, including procedures and functions.

Why is repetition used in programming? Repetition in programming is used to make a task happen over and over again without writing the same code multiple times, which helps save time and makes the program easier to manage.

What loops can be used in programming? In programming, you can use loops like **for loops** to repeat a task a set number of times and **while loops** to keep doing something until a condition is met.

Why are triggers used in programming? Triggers are special instructions that make something happen automatically when a specific event occurs, like pressing a button or receiving a message.

How can we check our programs are effective and debugged? We can check our programs are effective and debugged by testing them to see if they work correctly and fixing any mistakes we find.

What is rubber duck programming? Rubber duck programming is a method used by programmers, whereby you explain your code out loud to a rubber duck (or any object) to help you find mistakes and understand it better.

Why do we move from block based to text based programming? How does beginning with block based programming help us? We move from block-based to text-based programming to gain more control and flexibility in writing complex code. Starting with block-based programming helps us understand fundamental programming concepts visually using premade code segments (blocks), making it easier to transition to text-based coding later on. Text based programming means we can make our own blocks (subroutines) to carry out any task required using code.

Do you want to know more?

https://en.scratch-wiki.info/wiki/Animating_a_Sprite

https://thestempedia.com/tutorials/scratch-animations/

https://www.youtube.com/watch?v=mLzYpXcbv84

http://codedstem.org/content/cs_for_all/lectures/AdvancedProgrammingWithScratch.pdf

https://scratch.mit.edu/studios/3973114/

https://www.youtube.com/watch?v=Pe2fXVFrIDk

https://www.create-learn.us/blog/advanced-coding-with-scratch-a-how-to-for-kids-and-parents/

https://www.youtube.com/watch?v=K0T7zuxEIgw

Knowledge organiser - Computing - Year 7 - Summer 2

National Curriculum Strand:

- [x] Computer Science
- [ ] Digital Literacy
- [ ] Information Technology

Key Computing concepts

Flowcharts

Control Systems

Careers that might use this knowledge

Games designer

Air conditioning engineer

Marketing and sales

Progression path

I can develop responsive systems of control and represent these clearly using flowcharts

I can develop sub-routines (procedures) within my system to improve its effectiveness

I can include loops and subroutines in my systems for efficiency

I can build simple computer systems using control software based on flowcharts

I recognise that flowchart symbols represent steps in a computer system

I recognise situations where computers control systems through sensors or triggers (Traffic lights, lighthouses etc.)

I know that systems can be controlled by computers

Key Vocabulary

Algorithm	Sensor	Input	Flowchart
Control software	Process	Output	Process
Loop	Subroutine	Mimic	Control system
Decision	Condition	Terminator	Connector

Prior knowledge:

Pupils understand that algorithms are sets of step-by-step instructions used to solve problems or complete tasks. They have explored simple algorithms through practical activities, such as giving instructions for everyday tasks (e.g., making a sandwich or brushing teeth). In KS2, they have applied algorithms in basic coding environments (recognising how computers follow instructions precisely.

Pupils should have been introduced to flowcharts as a visual way to represent algorithms.

They know what simulations are.

Leading to:

As pupils progress through KS3 and KS4, their ability to think algorithmically becomes crucial in Computer Science, IT/Digital Skills, and Business Studies. In KS3, they advance from basic flowcharts to complex algorithms, including loops and conditionals, while learning pseudocode to structure logic before coding. In KS4 IT and Digital Skills, flowcharts help map digital processes, supporting automation, troubleshooting, and efficiency. In Business Studies, they illustrate decision-making models, marketing strategies, and workflow optimisation. Mastering flowcharts and pseudocode strengthens structured problem-solving skills, preparing students for careers in technology, analytics, and management.

Key Questions pupils should have the knowledge to answer

What are flowchart symbols? A graphical way of representing algorithms. symbols are graphical representations of different steps in an algorithm. They help programmers and engineers visualise processes before implementing them in code or systems. Common symbols include: Oval (Start/End) – Marks the beginning or end of a flowchart, rectangle (Process) – Represents an action or task, diamond (Decision) – Indicates a choice, often leading to different paths, and parallelogram (Input/Output) – Shows data being entered or displayed.

What is control software? This is the use of a computer to monitor and control an external process. Input sensors are used by the computer to monitor the various parts of a process that it is controlling. Before a system can be controlled, a human programmer must write a control program, ensuring the computer responds appropriately to sensor data. Control software is found in everyday systems like traffic lights, smart home devices, and industrial automation.

What is a mimic? A mimic represents a real life system or situation online. This is because of cost or safety of experimenting with the real situation. Mimics help refine designs, identify issues, and make improvements before applying solutions to physical systems.

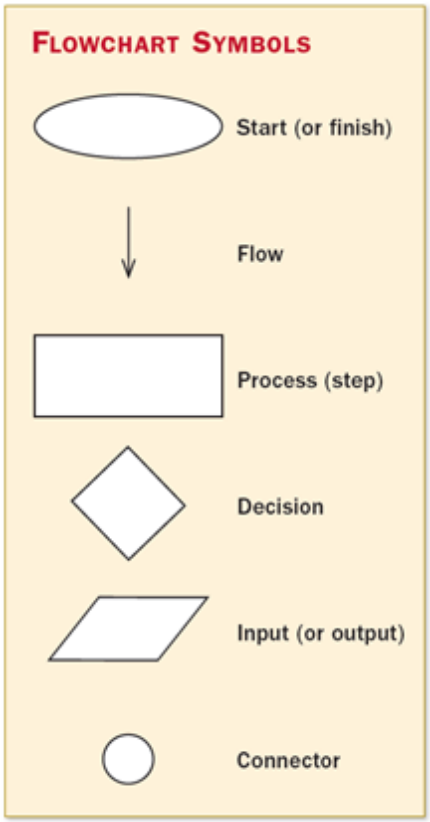
What is an input in a flowchart? An input in a flowchart is data or a signal received by a system, which then affects how the process continues. Inputs can come from a user, such as typing a value into a program, or from a sensor, such as detecting temperature or movement. In a flowchart, inputs are represented by parallelogram symbols.

What is an output in a flowchart? An output is the result or action produced by a system after processing an input. This could be displaying a message on a screen, turning on a light, or activating a motor. In flowcharts, outputs are also represented by parallelograms and show what happens once the system completes a step.

What is a loop in a flowchart? A loop in a flowchart represents a repeated process that continues until a certain condition is met. Loops are useful in automation where tasks need to happen multiple times, such as traffic lights cycling through colours or a washing machine running until a timer finishes.

What is selection in a flowchart? Selection is a process in which a decision is made based on conditions. It allows a flowchart to follow different paths depending on an input or a rule. Selection is represented by a **diamond-shaped decision box** in a flowchart, where a condition is checked before determining the next step. For example, a flowchart controlling a smart light might include the condition **"Is it dark outside?"** If the answer is **Yes**, the light turns on; if the answer is **No**, it stays off. Conditions are used in selection to allow **if statements** and **decision-making processes** within a system.

Why is selection important in control systems? Selection helps systems make intelligent choices rather than following a single sequence. In control software, selection allows devices to react to **real-world changes**, such as **adjusting the temperature in a greenhouse** or **activating traffic lights when a button is pressed**. Without selection, every process would follow a fixed path regardless of inputs.



Do you want to know more?

https://revisionworld.com/gcse-revision/ict/software/computer-control-software

https://www.bbc.co.uk/bitesize/guides/zfjsgk7/revision/2

https://www.gliffy.com/blog/guide-to-flowchart-symbols

https://www.bbc.co.uk/bitesize/topics/zs7s4wx/articles/zxjsfg8

https://kids.kiddle.co/Flow_chart

https://www.youtube.com/watch?v=Icyj4U8mIjQ

https://www.youtube.com/watch?v=nlotZCvL9RU

https://www.youtube.com/watch?v=kUt0nS0yMtM