COMPUTING SCIENCE

Exemplified Learning

LIVE
LEARN
WORK
INVEST
VISIT

COMPUTING SCIENCE – Exemplified Learning

Learning, Teaching and Assessment

Learners are at the heart of effective learning, teaching and assessment approaches as exemplified in the Learning, Teaching and Assessment Cycle. Staff in all sectors should regularly engage in moderation to ensure a shared understanding of standards and expectations. This will be ongoing and not seen as an end verification of results, but instead a continuous ongoing professional dialogue from the planning stages of effective learning, teaching and assessment. Learners need to understand the purpose of the learning as well as the continuous nature of learning.

The benchmarks are designed to support teachers' professional judgement of progress towards and achievement of a level. It is important that a collective understanding of what achievement looks like is achieved at school level, locally and nationally. This ensures that judgements are consistent, robust and reliable. This is achieved through effective moderation of planning learning, teaching and assessment both within and out with an establishment.

Usage Guide

- The Exemplified Learning activities are designed to be flexible and build on effective practice, highlighting knowledge and skills and exemplifying possible pace and progression pathways through a level. These are designed to be used in conjunction with the NL Progression pathways. Each theme has the core learning at the top followed by the suggested activities.
- Whilst aspects of particular curricular areas require sequential learning and teaching, this is not applicable to all areas and organisers and therefore the exemplified learning should be used flexibility to meet learners needs.
- Each pathway is split across three levels to highlight progress through a level. Learning is not a linear process. Learners can and should move within and across progression levels to support pace and challenge.
- Under each theme, at the top of the suggested activities section there is an overarching box, this is for activities that can be used and revisited across the level.
- Higher Order Thinking Skills are highlighted in bold and can be used to support staff to form Learning Intentions and Success Criteria.
- Progression pathways are not resource driven to allow staff the autonomy select the most appropriate resources to best suit learners needs and interests.
- Exemplified learning activities are designed to be flexible and to allow establishments, where applicable autonomy to determine the most appropriate.

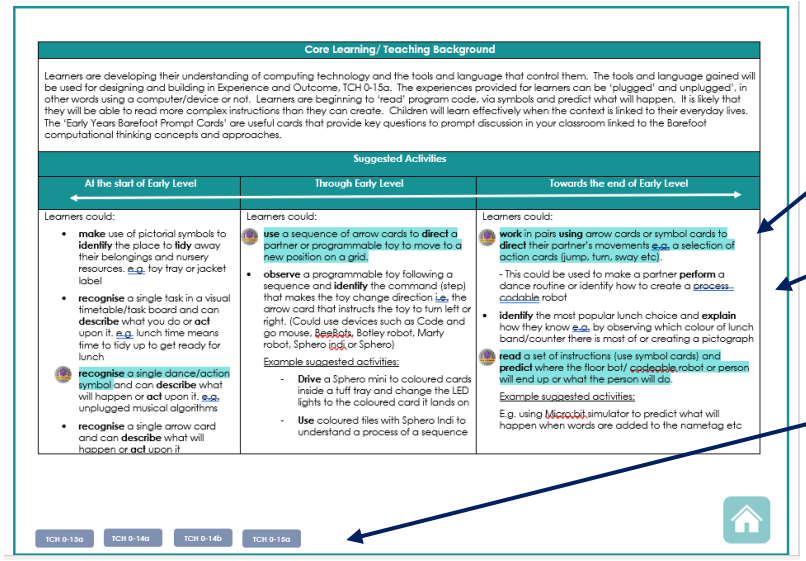
COMPUTING SCIENCE – Exemplified Learning

Key for using this document

Note - Not every suggested activity needs to be completed for children to have achieved the benchmark for that outcome, these are various suggestions to support the teaching and learning of these core learning outcomes

Top Tips :

- Icon buttons at the bottom of each theme link off to the Education and Families Sharepoint or Virtual Classroom to support the implementation of these planners e.g. Digital CLPL Toolkit to access recorded webinars.
- Exemplified learning activities are designed to be flexible and to allow establishments, to determine the most appropriate tool to use e.g. Most activities provide an iPad suggestion and an alternative laptop/desktop equivalent.
- Throughout this document there will be various hyperlinks to guides, CLPL or lesson examples. These will be in blue text with an underline. Simply click these hyperlinks to access the material.



Look out for our Tech Tuesday logo. Click the light teal highlight to take you to the Tech Tuesday video lesson. Click on light purple highlights to visit a Quick Guide video.

Look out for blue text, that is underlined. This is hyperlinked to various documentation, lesson ideas, 'how to guides' to help support you in teaching the concept

Click these navy blue buttons if you are looking for a specific organiser e.g. TCH 0-14b

Click the green home button to come back to the guidance

- All **Tech Tuesday** recorded lessons and teaching PDF's are linked specifically to the Exemplified Learning bullet points. We have highlighted in light teal to the exact Tech Tuesday lesson which covers the suggested activity to make this exemplified learning document a 'one-stop- shop'.

Organiser - Understanding the World through Computational Thinking

Second Level

Experiences and Outcomes covered	I **understand** the operation of a process and its outcome. I can **structure** related items of information	TCH 2-13a

Core Learning

At the start of Second Level	Through Second Level	Towards the end of Second Level
I can:	I can:	I can:
• **Identify** and **follow** steps in a sequence of single step and multiple parallel step activities.	• **Compare** how the steps in a sequence within single step and multiple parallel step activities, affect the outcome.	• **Compare** and **demonstrate** an understanding of how the steps within a single step sequence and a multiple parallel step process, affect the outcome.
Explore examples of algorithms/instructions which includes fixed and conditional repetition processes e.g. board games.	• **Discuss** and **decide** whether an algorithm/instruction includes fixed and conditional repetition processes using examples of those which do and do not.	• **Identify** conditional or fixed repetition within an algorithm/instruction.
Explore a variety of activities that involve following a process (algorithms) but can have more than one outcome, e.g. Rock, Paper, Scissors.	• **Identifies** when a process is not predictable e.g. Toss a Coin, Bingo, Snap or Higher/Lower, board games which uses dice or spinners.	• **Identify** the random element in a variety of processes when it is not possible to **predict** the outcome.
• **Explore** and **identify** ways to structure information e.g. family tree, flow charts, data tables, spreadsheet, sorting network etc.	• **Make** decisions, with support, about how to **arrange** and **structure** related items of information e.g. flow chart for making lunch choices.	**Make** decisions to **arrange** & **structure** related items of information e.g. build a family tree, school hierarchy - pupils, teachers, admin staff, combination graphs etc.
• **Use** sorting algorithms to **sort** everyday objects i.e books alphabetically, sorting pupils into height, age order.	• **Explore** and **use** more complex ways in which information is sorted & stored e.g. Top Trump cards, card games – solitaire, round the clock.	**Explore** and **use** more complex sorting algorithms, **explaining** the logical thinking used within the sorting process, e.g. Bubble Sorting.

Benchmarks covered

- **Compares** activities consisting of a single sequence of steps with those consisting of multiple parallel steps, for example, making tomato sauce and cooking pasta to be served at the same time.
- **Identifies** algorithms/instructions that include repeated groups of instructions a fixed number of times and/or loops until a condition is met.
- **Identifies** when a process is not predictable because it has a random element for example, a

First Level

Third Level

TCH 2-13a TCH 2-14a TCH 2-14b TCH 2-15a

			board game which uses dice. ○ **Structures** related items of information for example, a family tree (MNU 2-20b). ○ **Uses** a recognised set of instructions/an algorithm to **sort** real world objects for examples, books in a library or trading cards.

Digital CLPL Toolkit

NL Virtual Classroom

NL Digital School Documentation

TCH 2-13a TCH 2-14a TCH 2-14b TCH 2-15a

Core Learning/ Teaching Background

Learners are building on from their prior knowledge of **sequence**, **selection** and **repetition**. Learners will explore parallel processes i.e. two activities working alongside and/or interacting with each other. Learners will compare everyday activities with single sequence steps to multiple parallel steps in order to identify and describe the properties of parallel processes. They will identify when a repeated set of steps is fixed (it loops a known number of times) or conditional (it loops until a condition is met). Learners will begin to be able to understand when an outcome is able to be predicted or not, i.e. there is an element of randomness. This learning can be achieved through play and taking part in games e.g. playground games, dance and board games. Research suggests children who are introduced to 'unplugged' activities before switching to 'plugged' gain stronger computational skills and are more motivated, (Kunkeler, T - Unplugged for Better Computational Thinking).

At the start of Second Level	Through Second Level	Towards the end of Second Level
Learners could: • **identify** and **follow** the steps in a sequence of a single step activity, e.g. making toast; and a multiple step activity, e.g. making toast and a cup of tea. • **explore** an online game; such as The Beetle Game and **identify** two parallel processes. • **explore** examples of board games which have fixed and conditional repetition processes. For example: - Fixed – If you land on a specific space, you miss 3 turns. - Conditional – Keep rolling dice until you roll a 6 to start game (Ludo).	Learners could: • **compare** the steps in a sequence of a single step activity, e.g. making toast; with a multiple step activity, e.g. making toast, scrambled eggs and a cup of tea. Learners **discuss** timings and the sequence of steps taken in order to ensure all are ready simultaneously. • **explore** online games, such as Clever Crab and Find A Key and **compare** the single step and two parallel processes. • **explore** and **discuss** examples of board games which have randomness, fixed and conditional repetition processes and games that do not. • **explore** and **discuss** examples with and without fixed and conditional repetition (loops) blocks within a programme of script. Then **identify** which scripts include repetition and which do not. • **identify** when a process is not predictable by **exploring** a variety of games, e.g. Toss a Coin,	Learners could: • **compare** the steps in a sequence of a single step activity, with a multiple step activity. Learners **demonstrate** an understanding of timings and the sequence of steps taken in order to ensure all are ready simultaneously. • **explore** online games, such as Scribbling Dog and vs Pacman's role then **compare** and **explain** the roles of the steps within the single and multiple parallel processes of the games. • **identify** examples fixed and conditional repetition processes within various board games and explain what each step is doing. Learners **create** their own game which includes either fixed and conditional repetition processes. • **identify** examples of repetition (loops) blocks within a programme of script and **determine** if it is conditional or fixed, then **explain** what the block is doing in the programme.

TCH 2-13a | TCH 2-14a | TCH 2-14b | TCH 2-15a

• **explore** fixed and conditional repetition (loops) blocks within a programme of script. **explore** a variety of games that involve following a process (algorithms) to play, but can have more than one outcome, e.g. Rock, Paper Scissors. • **explore** and **identify** ways to structure their own family tree, **demonstrate** how to represent various relationships between family members. Learners can **explore** other ways to structure information, e.g using data tables, spreadsheets, sorting networks. • **use** simple sorting rules/algorithms to **sort** everyday objects, e.g. sorting books alphabetically, pupils into height, age order etc.	Bingo, Snap or Higher/Lower and board games that use dice or spinners • **make** decisions, with support, to **arrange** and **structure** related items of information, e.g. flow chart for making lunch choices, decision tree for classifying animals. • **explore** and **use** more complex ways in which information is sorted, **discussing** what processes they went through to get the desired outcome, e.g. sorting Top Trumps cards by their category values, playing Solitaire or Round the Clock to sort cards into suits. Example Suggested Activities Learners **explore** Artificial Intelligence by teaching a machine to recognise different words and phrases to create a weather detector project • Learners explore online sorting algorithm games.	• Learners **identify** the random element in a process when it is not possible to **predict** the outcome by **exploring** a variety of games, e.g. using a dice in Ludo, using a number generator in Bingo. Learners **make** decisions to **arrange** and **structure** related items of information, e.g. build a family tree, a chart showing school hierarchy - pupils, teachers, admin staff. • Learners **explore** and **use** more complex sorting algorithms, **explaining** the logical thinking used within the sorting process, e.g. bubble sort, quick sort, selection sort. Learners are assigned a number, on their whiteboard, from 1 to 10 and placed in a line randomly. Learners **sort** themselves in to ascending or descending order **using** any of the sorting algorithms learned, e.g. bubble sort: - Learner 1 = L1 - L1 and L2 step forward and face each other, if L1 is larger than L2 they swap places. Then L2 and L3 step forward and face each other, if L2 is larger than L3 they swap places. This continues up the line and back down as many times to get the numbers in the correct order.

TCH 2-13a TCH 2-14a TCH 2-14b TCH 2-15a

Key Vocabulary	Cross Curricular Links
Algorithms, instructions, evaluation, collaboration, repetition, fixed, conditional, debugging, creating, logic, tinkering, sorting, processes, parallel, random	• I have carried out investigations and surveys, devising and using a variety of methods to gather information and have worked with others to collate, organise and communicate the results in an appropriate way. **MNU 2-20b** • I can conduct simple experiments involving chance and communicate my predictions and findings using the vocabulary of probability **MNU 2-22a** • When listening and talking with others for different purposes, I can share information, experiences and opinions ·explain processes and ideas, identify issues raised and summarise main points or findings ·clarify points by asking questions or by asking others to say more. **LIT 2-09**

Computational Thinking concepts and approaches	Additional/Extension activities	Teacher CLPL Materials
• Logic • Algorithms • Decomposition • Pattern • Abstraction • Evaluation • Tinkering • Creating • Debugging • Persevering • Collaborating **Barefoot concepts and approaches**	• What is an algorithm? - BBC Bitesize • Jam Sandwich Algorithm | code-it supported by HIAS, Hampshire Inspection and Advisory Service • Flowcharts & repetition | micro:bit (microbit.org) • www.spen.org.uk/mymedia/files/resource_pdfs/Youth Scotland - Board Game Toolkit.pdf • A Bit of a Dicey Problem (maths.org) • Math Bingo Game | Super Cool Math Workouts | Toy Theater • Sorting networks - CS Unplugged • Classifying living things with CS	• Teach-CS-Oct18.pdf (teachcs.scot) • Computing Science by digilearn.scot (glowscotland.org.uk)

TCH 2-13a TCH 2-14a TCH 2-14b TCH 2-15a

Organiser - Understanding and analysing computing technology

Second Level

Experiences and Outcomes covered	I can **explain** core programming language concepts in appropriate technical language.	TCH 2-14a

Core Learning			Benchmarks covered
At the start of Second Level	**Through Second Level**	**Towards the end of Second Level**	
I can: • **Discuss** variables in everyday life and **identify** variable coding blocks in a visual programming language. • **Discuss** conditional repetition in everyday life and **identify** conditional repetition coding blocks in a visual programming language. **Discuss** what a program in a visual programming language will do when it runs and **explain**, with support, how the properties of the object change with each instruction. • **Predict** what a program in a visual programming language will do when two parallel processes (activities) run e.g. when two sprites	I can: • **Explore** and **explain** how variable coding blocks are used in a visual programming language e.g. timers and score counters. **Explore** and **explain** how conditional repetition coding blocks are used in a visual programming language. **Discuss** what a more complex program in a visual programming language will do when it runs and **explain**, with support, how the properties of the object change with each instruction. **Predict** what a program in a visual programming language will do when parallel processes (activities) run and interact e.g. when two sprites run simultaneously but can communicate and/or affect one another.	I can: **Explain** the meaning and functions of individual coding blocks including variables and conditional repetition. • **Explain** and **predict** what a complex program in a visual programming language will do when it runs and how the properties of the object change with each instruction. • **Explain** and **predict**, using visual programming language, how parallel processes (activities) interact e.g. how selection and sensing blocks are used to detect another sprite and looks blocks are used to 'hide' a sprite.	○ **Explains** the meaning of individual instructions (including variables and conditional repetition) in a visual programming language ○ **Predicts** what a complete program in a visual programming language will do when it runs, including how the properties of objects for example, position, direction and appearance change as the program runs through each instruction. ○ **Explains** and **predicts** how parallel activities interact.

First Level

Third Level

TCH 2-13a | TCH 2-14a | TCH 2-14b | TCH 2-15a

run simultaneously but do not affect each other.			

Digital CLPL Toolkit

NL Virtual Classroom

NL Digital School Documentation

Progressive Skills Map

Keyboard and mouse control

CS Quick Guides

TCH 2-13a TCH 2-14a TCH 2-14b TCH 2-15a

Core Learning/ Teaching Background

Learners are developing their understanding of computing technology and the tools and language that control them. The tools and language gained will be used for designing, building and testing in Experience and Outcome, TCH 2-15a. The experiences provided for learners can be 'plugged' and unplugged', in other words using a computer/device or not. Learners are developing their ability to read and understand representations of processes in a programming language. They will identify examples of sequential, selective, repetitive, parallel control structures and use of variables. They can select and describe individual structures and well as predict the outcome of a program. Research suggests learning to read code is more effective before learning to write code, furthermore learners will be able to read more difficult code than they are able to write.

At the start of Second Level	Through Second Level	Towards the end of Second Level
Learners could: • **use** a physical box(es) and counters to keep score of a game, helping them to understand that a score is temporary data which can change i.e. variable(s). • be introduced to variables through an unplugged activity where they keep score in a game. Learners can then go on to discuss other variables in everyday life. • **explore** conditional repetition in everyday life. They can explore the concept and suggest examples e.g. automatic car window wipers will go back and forth until it stops raining. Learners begin to understand that conditional repetition is used in programming to repeat an action until a condition is met e.g. a sprite	• Learners could: • **explore** how to **use** variable coding blocks in a visual programming language. Learners use variables to keep score in a previously created quiz. • **explore** how to use conditional and repetition coding blocks in a visual programming language. Learners explore using conditional repetition in game-based setting. • **use** more complex programs containing conditional repetition and variables to **predict**, **run** and **investigate** what will happen to the properties of the sprites when run. • be given a complete programme which includes parallel processes that **interact**. They will attempt to **predict** what will happen when script runs then **run** the program **check** if their prediction is correct.	Learners could: • **explain** the appropriate coding blocks needed to achieve the desired outcome, for example what blocks are needed to collect points, timer, record temperature, sound etc. • **explain** the appropriate coding blocks needed to achieve the desired outcome, for example what blocks are needed to keep a sprite in continual motion until a condition is met e.g. touches the edge of a screen. • presented with different scripts which achieve the same outcome however one script has an error. They **identify** the script with the error and **explore** the benefits of using efficient scripts i.e. those that use repetition have less room for error and are more efficient. • **predict** using visual programming language, how parallel processes (activities) interact. They can explain how and where the scripts interact.

TCH 2-13a TCH 2-14a TCH 2-14b TCH 2-15a

continues to move until it touches another sprite. • Learners can explore conditional repetition further with playing card game. • Learners correctly **match** the visual programming block to the action it performs, including blocks for variables and conditional repetition. Learners use simple programs provided to them to **predict, run** and **investigate** what will happen to the properties of the sprites when run. • Learners are introduced to the concept of parallel scripts. Learners are given a complete simple programme which includes parallel processes which do not interact. Learners will attempt to **predict** what will happen when script runs then **run** the program to **check** if their prediction is correct.		

TCH 2-13a TCH 2-14a TCH 2-14b TCH 2-15a

Key Vocabulary	Cross Curricular Links
background, coding blocks, conditional repetition, efficient, error, loops, fixed repetition, parallel process, predict, repetition, run, script, selection, sequence, sprite, tinker, visual program	• I can show my understanding of how the number line extends to include numbers less than zero and have investigated how these numbers occur and are used. **MNU 2-04a** • Through practical activities which include the use of technology, I have developed my understanding of the link between compass points and angles and can describe, follow and record directions, routes and journeys using appropriate vocabulary. **MNU 2-17a** • I can use my knowledge of the coordinate system to plot and describe the location of a point on a grid. **MNU 2-18a** • When I engage with others, I can respond in ways appropriate to my role, show that I value others' contributions and use these to build on thinking. **LIT 2-02a**

Computational Thinking concepts and approaches	Computer Science Concepts and Approaches	Additional/Extension activities	Teacher CLPL Materials
• Logic • Algorithms • Decomposition • Patterns • Abstraction • Evaluation • Tinkering • Creating • Debugging • Persevering • Collaborating **Barefoot concepts and approaches**	• Variables • Sequences • Programming • Repetition	• All Blocks of Scratch • Scratch Maths Quiz Selection | Resources | Barefoot (barefootcomputing.org) • Maths Quiz Variables | Resources | Barefoot Computing • Creative Computing Curriculum • Scratch Tinkering • BBC Bitesize Computing Science and ICT	• Teach Computing Science A Guide for Primary and Early Years Practitioners • DigiLearn Scot Computing Science Blog • Scratch for Educators • Barefoot Computing • Get Started with BBC Micro:bit • Code.org Careers in Computing Science

TCH 2-13a TCH 2-14a TCH 2-14b TCH 2-15a

Organiser - Understanding and analysing computing technology

Second Level

Experiences and Outcomes covered	I understand how information is stored and how key components of computing technology connect and interact through networks.	TCH 2-14b

Core Learning

At the start of Second Level

I can:

- **Explore** how binary code is formed from 0s and 1s and how it is used to represent data.
- **Explain** the basic functions of the various internal parts of a computer and **explore** their role in storing and processing information e.g. the processor is the computer brain etc.
- **Explore**, in simple terms, what a network is and how it stores and transfers data to communicate. **Demonstrate** an understanding that there are different types of networks responsible for this.
- **Demonstrate** an awareness of a network's ability to communicate and **share** information online e.g. internet

Through Second Level

I can:

- **Explore** and **investigate**, in simple terms, how all data (numbers, text and graphics) is translated and represented by binary.
- **Use** the correct terminology to **describe** parts of the computer e.g. CPU, motherboard, hard drive, USB drive, SD cards etc.
- **Demonstrate** a basic understanding of processing and memory components on different devices.

  Demonstrate an understanding of how information is stored on school networks (wired and wireless, server, router) and **explain** the journey that data takes e.g. when sharing from My Files (OneDrive) to a SharePoint in Glow.

Towards the end of Second Level

I can:

- **Demonstrate** an understanding of how computers hold all data in binary form.
- **Describe** the purpose of and the relationship between the processor, memory and storage and **apply** this knowledge to **discuss** different devices.
- **Explain**, in simple terms, how data is transferred between different networks and servers e.g. LAN and WAN.

Benchmarks covered

- Demonstrates an understanding that all computer data is represented in binary for example, numbers, text, black and white graphics.
- Describes the purpose of the processor, memory and storage and the relationship between them.
- Demonstrates an understanding of how networks are connected and used to communicate and share information, for example the internet.

First Level

Third Level

TCH 2-13a | TCH 2-14a | TCH 2-14b | TCH 2-15a

searches, email, collaborative work in GLOW.			

Core Learning/ Teaching Background

Computers are machines that carry out well-defined processes that manipulate information. A computer system does not do these things spontaneously, it is told precisely what to do. Learners will continue to explore how different computer systems work to handle, store and share information; begin to gain an understanding of the basics of how computer networks operate - how they transfer information between computers/devices using a 'universal' language called binary code; understand how software, such as apps or web browsers, store and transmit information too.
Throughout this outcome, pupils will should be regularly encouraged to make links between their learning and what is going on inside the everyday devices we use.

All computers do the same four tasks - **input** information from the physical world, **store** and **process** the information then **output** information back into the physical world. Different parts of the computer are responsible for carrying out these tasks. The more complicated the task and the greater the input and output, the more processing power and memory a computer needs. Learners should come to realise that we need outputs to show us that the computer has followed the instructions it was given and has completed the task.

Computers today use digits to represent information - that's why they're called digital systems. The simplest and most common way to represent digits is the binary number system, with just two digits (usually written as 0 and 1). Computers store text, drawings, photographs and other information using only numbers but sometimes the files are quite large and require a lot of storage space. As computers have a limited amount of storage space, these activities help learners to explore how digital information can be 'shrunk' (**compressed**) to take up less memory space and make it more transportable too.
As we use computers more and more, we want them to process information as quickly as possible. One way to increase the speed of a computer is to write programs that use fewer computational steps.

Links to TCH 2-05a benchmark: *Gives examples of how our changing lifestyles have impacted on product design*. (Components, peripherals, interface and processing)

Core Learning

At the start of Second Level	Through Second Level	Towards the end of Second Level

TCH 2-13a | TCH 2-14a | TCH 2-14b | TCH 2-15a

Introduction to binary code • Learners will be **introduced** to the basics of Binary Code in a lesson that shows how computers represent words and numbers using just 0s and 1s. There are several activities contained within this linked lesson that reinforce the learning of Binary Code. Computer Systems and Information • Learners **undertake** a lesson to determine what types of inputs and outputs a range of computing devices use. Groups are **assigned** a different device and list its inputs and outputs. Throughout the lesson the teacher records inputs and outputs that are identified on a T-Chart at the front of the room. Learners then **examine** common activities they do on a computing device and **select** the inputs and outputs used for that activity from the chart. • Learners will **deepen** their understanding of how the computer unit, itself, works and that the central processing unit (CPU) is responsible of what happens – it is, essentially, the brains of the computer. Learning can be centred around discussion and video aids can reinforce (or may	Introduction to binary code • Learners will **complete** a lesson that encodes letters into binary, decodes binary back in to letters and relates the idea of storing initials on a bracelet to the idea of storing information in a computer. Additional teacher guidance video: Unplugged Lesson in Action - Binary Bracelets - YouTube Computer Systems and Information • Learners **watch** a video about the main parts of a computer and then are given the chance to **tinker** with various redundant devices and components. As they **tinker** they locate the main components, **compare** how their inputs and outputs differ and **discuss** how the devices may have changed over time. If no redundant hardware is available to strip back, picture cards could be used or the following video stimuli: What does what in your computer? Computer parts Explained or Computer Basics: Inside a Computer • Learners **take part** in an iPad simulation activity to get a basic sense of how computers work. Each student **takes on** the role of a different part of a computer and they work in **groups** to run a simple program. • Learners continue to **develop** their knowledge of computer memory,	Introduction to binary code • Learners will **use** binary coding to **represent** pathways, through a series of "high" and "low" choices, to **represent** colour and images. Learners will then **explore** how electronic devices use binary signals to determine what types of images to create using pixels and go on to **create** their own pixel art using binary code. Use Binary to Make Pictures activity (Pages 12-20) Computer Systems and Information • Learners will **complete** an investigation in which they are challenged to **create** a Scratch program that uses the input from a device: Investigating Inputs Activity | Resources | Barefoot Computing • Learners **take part** in a computer simulation to get a basic sense of how computers work. Each learner **takes on** the role of a different computer part and **works** in a group to simulate the running of a simple program. • Learners will expand their learning and explore examples of control programs. They will create a sound monitor for their classroom with Scratch or Micro:bits. Learners can then use this learning to explore and discuss other places they see sensor control in place. • Learners **share** prior knowledge and continue to **develop** understanding of computer memory, processing and storage through discussion. Video resource(s) useful as discussion prompts - Memory & Storage: Crash Course Computer Science #19 - YouTube • Learners **take part** in a 'magic trick' (led by the teacher) to **show** how to **detect** when data has

TCH 2-13a TCH 2-14a TCH 2-14b TCH 2-15a

introduce) key vocabulary and learning. Informational videos to introduce: • Inside your computer - Bettina Bair • Computer Basics: Inside a Computer • Khan Academy and Code.org | CPU, Memory, Input & Output - YouTube • Learners, with support, will **take part** in discussions based around these key points: - Memory – short term and long term, volatile or non-volatile - Primary storage - Secondary storage • Video resource(s) can be used to prompt the discussion Evolution of storage device - YouTube	processing and storage through discussion. Video resource(s) useful as discussion prompts: • Evolution Of Data Storage Devices - YouTube • How computer memory works - Kanawat Senanan - YouTube • Learners will **discover** that, by coding data before it is stored, and decoding it when it is retrieved, the computer can store more data, or send it faster through the Internet. Learners will **examine** letter patterns in simple poems/rhymes and **learn** how computers spot these and are able to compress (combine) the text data to make it smaller in size (links to Binary). Text Compression | Classic CS Unplugged	been corrupted, and to correct it. Learners will then **learn** the trick and go on to work in small groups to investigate for themselves. Links to Binary code are made here too. Learners will then **look** at a real-life examples of error detection through ISBN book numbers. Error Detection | Classic CS Unplugged • Learners can further **develop** their understanding of information/data compression by **completing** extension activities from the resource: Text Compression | Classic CS Unplugged.
Learners will **demonstrate** their understanding and learning of Computer Systems and Information through Digital Literacy based on TCH 2-01a i.e. saving, storing and transferring files/info via on computer drive, cloud drive etc.		
Understanding Networks • Learners will **go** on a hunt around their school to **discover**, and **map** the location of, devices connected to their school's network. They then **learn** about the role of each device by either **conducting** web-based research or **using** video resources such as the Khan Academy playlist could be used to	Understanding Networks • Learners **discuss** the different types of 'networks' they can find beyond school and at home. Examples of what to look for - modems, routers, energy meters, voice-activated digital assistants, smoke/heat detectors, video doorbells etc. Possible discussion prompt: Computer Networks: Crash Course Computer Science #28 - YouTube.	Understanding Networks • Learners will be shown that the internet is a vast network of computers and other devices connected across the world, as they **explore** the difference between the internet and the World Wide Web (WWW). Learners are **assigned** roles as different digital devices in a human model of the internet and **learn** how it provides access to the WWW as they are asked to **pass** data between them: Modelling the Internet | Resources | Barefoot Computing

TCH 2-13a TCH 2-14a TCH 2-14b TCH 2-15a

stimulate discussion. Examples of what to look for - computing devices, modems, routers, BT boxes, CO_2 monitors etc: Network Hunt Activity | Resources | Barefoot Computing • Learners will **explore** the problem of when there are lots of people using one resource at the same time, that there is a possibility of "deadlock". They will **try** to understand that a way of working cooperatively is needed to avoid this happening: Routing and Deadlock | Classic CS Unplugged – The Orange Game • Learners will **explore** the basics of how search engines use web crawlers to index the World Wide Web (WWW). They will **act** like web crawlers themselves, **indexing** a very small portion of the WWW, and they then **use** this index to respond to search queries: Selecting Search Activity | Resources | Barefoot Computing	• Learners, with support, **explore** sorting networks which do several sorting comparisons at the same time. This is team activity that demonstrates an approach to parallel sorting. It can be done on paper, but is ideally done on a large scale: - Sorting networks - CS Unplugged - Reinforcing numeracy through a Sorting Network - CS Unplugged • Learners will **learn** about some of the main factors which influence how a search engine ranks a web page. They'll **create** paper-based 'web pages' in groups on a current topic they are studying. Learners then **discover** how their web pages would rank when searching for keywords relating to their content: Ranking Search Activity | Resources | Barefoot Computing	• Learners further **explore** sorting networks by **applying** a range of variations to the data going through the network e.g. numbers, alphabetical, musical pitch etc: - Sorting networks - CS Unplugged - Investigating variations using the Sorting Network - CS Unplugged • Learners **complete** a task that makes use of and **demonstrates** their learning about the WWW, internet searches and ranking. Teachers can provide a stimulating and familiar context for learners e.g. compiling a Transition to High School help guide for peers, complete a WW2 research task. Learners will **explore** whether the top suggestions on the search list are more relevant than those further down and **reflect** on effective use of key words when searching.
Learners will **demonstrate** their understanding and learning of communication and computer networks through Digital Literacy activities based on TCH 2-01a and TCH 2-02a.		

TCH 2-13a TCH 2-14a TCH 2-14b TCH 2-15a

Key Vocabulary	Cross Curricular Links
Device, computer system, components, peripherals, hardware, software, inputs, outputs, motherboard, processor, CPU, bits, memory, cache, primary storage (RAM, ROM, flash and virtual), secondary storage (disc, hard drive, USB drive), cloud drive, compress, network, WAN (Wide Area Network), LAN (Local Area Network), data, interact, connect, communicate, binary code, IP packet, ethernet, wi-fi, Bluetooth	• I have worked with others to explore, and present our findings on, how mathematics impacts on the world and the important part it has played in advances and inventions. **MTH 2-12a** • Having explored more complex number sequences, including well-known named number patterns, I can explain the rule used to generate the sequence, and apply it to extend the pattern. **MTH 2-13a**

Computational Thinking concepts and approaches	Computer Science Concepts and Approaches	Teacher CLPL Materials
• Logic • Algorithms • Decomposition • Patterns • Abstraction • Evaluation • Tinkering • Creating • Debugging • Persevering • Collaborating **Barefoot concepts and approaches**	• Computer Networks • Computer Systems • Control • Inputs • Outputs • Data • Internet Services • Search Technologies • Programming • Simulation • Variables	• How exactly does binary code work? - José Américo N L F de Freitas - YouTube • Binary | Interactive | Computing (advanced-ict.info) • Teaching Computing Systems and Networks to 5- to 11-year-olds - Teach Computing • Crash Course Computer Science Preview - YouTube Playlist • Processors Explained for Beginners | CPU's Explained for Beginners - YouTube • What is Digital Information - YouTube • Microsoft's virtual datacenter grounds 'the cloud' in reality | Innovation Stories • Gever Tulley: Life lessons through tinkering | TED Talk

TCH 2-13a TCH 2-14a TCH 2-14b TCH 2-15a

Additional/Extension activities

Binary:

- What is digital data? - BBC Bitesize
- CS Discoveries 2020-2021 | Data and Society (code.org)

Possible video aids to introduce the concept of Binary:

- Khan Academy and Code.org | Binary & Data - YouTube
- Binary Numbers for Kids | Convert Decimal to Binary | Computers for Kids - YouTube

- What is a CPU? What is Binary? - YouTube (Upper Second Level)

Computational Fairy Tales: Books (computationaltales.blogspot.com)
Each story serves to illustrate a computational concept, supplementing official instruction or motivating computer science concepts.

Networks:

- KS2 Computing - COMPUTING THEORY - 5. Computer networks
- The Story of Send - Emails & Environment | Online Animation on Google - YouTube

- How The Internet Works - YouTube Khan Academy and Code.org playlist

- Computers and the Internet | Code.org | Computing | Khan Academy

Key Stage 2 (teachcomputing.org) - Computing Systems and Networks units

Processing, Memory & Storage:

- What are input and output devices? - BBC Bitesize
- Inputs and outputs | micro:bit (microbit.org)
- Controlling physical systems - BBC Bitesize (extension knowledge)
- How Computers Work - YouTube Khan Academy and Code.org playlist
- Computer Processors Explained (Official Dell Tech Support) - YouTube (Pupil and teacher friendly)
- Workshop: Simulate computer - Children and Technology by Misha Leder (google.com)
- Central processing unit Facts for Kids (kiddle.co)
- CS Discoveries 2020-2021 | Problem Solving and Computing (code.org) (Chapter 2)
- Differences between RAM and ROM - YouTube
- RAM Vs. ROM | Animation - YouTube
- Classroom Resource: How Computers Store And Transmit Data (sciencefriday.com)

Organiser - Designing, building and testing computing solutions

Second Level

Experiences and Outcomes covered	I can **create**, **develop** and **evaluate** computing solutions in response to a design challenge	TCH 2-15a

Core Learning			Benchmarks covered
At the start of Second Level	**Through Second Level**	**Towards the end of Second Level**	
I can: **Create** a program specific to a design challenge, **using** a given set of coding blocks including fixed repetition. **Use** a visual programming language, with support, to **create** programs that begin to show use of variables and conditional repetition. **Recognise** that a solution from a previous problem can be **reused** to solve a new problem. **Explore** a given programmed solution and **determine** if there are any changes /improvements to be made e.g. bugs and discrepancies between task description and solution.	I can: **Create** a program specific to a design challenge, **using** a given set of coding blocks including variables and conditional repetition. **Identify** and **select** solutions from a previous problem and **reuse** these for a new problem. • **Evaluate** a programmed solution, **determining** any mismatches against the task description and **discuss** ways they can be fixed.	I can: **Create** a program specific to a design challenge **including** variables and conditional repetition. **Identify** and **use** aspects of previous solutions to solve new problems e.g. change the value of a score counter. • **Identify** any discrepancies between the task description and solution and **indicate** how to fix them. Eg. debug and solve problems in a Sphero Bolt program	**o Creates** programs in a visual programming language including variables and conditional repetition. **o Identifies** patterns in problem solving and **reuses** aspects of previous solutions appropriately, for example, **reuse** code for a timer, score counter or controlling arrow keys. **o Identifies** any mismatches between the task description and the programmed solution and **indicates** how to fix them.

First Level

Third Level

TCH 2-13a TCH 2-14a TCH 2-14b TCH 2-15a

Core Learning/ Teaching Background

Learners will apply their knowledge of Organiser 1 & 2 to create solutions to solve a design problem that includes variables such as a points system. They will follow along with guided lessons and create games using a block-based programming system, for example Scratch to explore fixed and conditional repetition, parallel programmes, selection and decision-making processes. Learners will start to build stronger problem-solving links and remix pieces of code to fit another purpose i.e. reuse code for a timer or score counter. Learners will continue to enhance their knowledge of debugging by creating their own bugged programmes and challenge their peers to identify and fix the bug.

At the start of Second Level	Through Second Level	Towards the end of Second Level
Learners could: • **Create** a game using a design brief challenge e.g. Ed Scot/NL Digital School Code Alongs • be given missions which include a problem that needs to be solved. Learners **use** their prior learning to **solve** the 'mission'. Missions can include: - Mission 1: Locator Beacon - Mission 2: Code Word - Mission 3: Password Generator - Mission 4: Hacking Passwords - Mission 5: Hide 'n' Seek **create** a program, with support, that **includes** basic variables and conditional repetition e.g. creating a game that allows a shark to catch fish and IF it touches the fish THEN the fish disappears, the score increases by one and the fish reappears randomly on screen.	Learners could: **create** a game using a design brief challenge e.g. Ed Scot/NL Digital School Code Alongs • Be given missions which include a problem that needs to be solved. Learners **use** their prior learning to **solve** the 'mission'. Missions can include: - Mission 1 – 5 plus… - Mission 6: Flight Simulator - Mission 7: Jumping Simulator - Mission 8: Catching Simulator • **discuss** and **identify** code from a previous problem can be **reused** to **solve** a new problem e.g. a timer counting down from sixty seconds could be reused to count up from zero to sixty seconds. • be given a completed program and a matching task description to **explore**. Learners begin to **evaluate** the completed program to **identify** any mismatches against the task description and begin to **discuss**, in pairs/small groups, ways any mismatches can	Learners could: **Create** a game using a design brief challenge e.g. Ed Scot/NL Digital School Code Alongs be given missions which include a problem that needs to be solved. Learners **use** their prior learning to **solve** the 'mission'. Missions can include: - Mission 1 – 5 plus… - Mission 6: Flight Simulator - Mission 7: Jumping Simulator - Mission 8: Catching Simulator - Mission 9: Chasing Simulator - Mission 10: Maze Escape **identify** small pieces of code from a previous problem that can be **reused** to **solve** a new problem e.g. using a score counter that increases when a condition is met can be reused to keep track of health points that decreases when a condition is met. • be given a completed program and a matching task description to **explore** independently. Learners begin to **evaluate** the completed program to **identify** any mismatches against the task description. Learners **present** their evaluation to a peer/small group and then **discuss** ways any mismatches can be fixed e.g. a programme

TCH 2-13a TCH 2-14a TCH 2-14b TCH 2-15a

- with support, begin to **recognise** that code from a previous problem can be **reused** to **solve** a new problem e.g. reuse arrow buttons code for another game that uses arrow buttons to move a sprite.

Learners are given a completed program and a matching task description to **explore**. They can **discuss** in pairs/small groups if the program meets the description and if not, **identify** what would be required to be edited e.g. a programme that will allow a sprite to catch falling objects doesn't work as the objects fall to the bottom of the screen.

Suggested Resources
Watch the assembly and then **explore** the micro:bit activities

Activity 1
Find out about careers in gaming

Tech for Gaming and eSports

Then complete the follow up micro:bit tasks:

Graphical Dice
Create a dice project that looks like a real die with patterns of dots instead of numbers.

be fixed e.g. a programme that will allow a sprite to chase and catch another sprite doesn't work as when the sprites touch nothing happens.

Suggested Resources
Watch the assembly and then **explore** the micro:bit activities

Activity 1
Find out about how technology supports people to help others

Tech to Help People

Then complete the follow up micro:bit tasks:

Compass North
Create a simple compass that will show you which way is North.

Spirit Level
Make a tool for making sure pictures, shelves or work surfaces are level. The new micro:bit's built-in speaker makes it easy to improve your spirit level with audio feedback.

Tell me a secret
Use the radio function to **send** a message and **answer** questions

Sound Insulation Meter
Use two BBC micro:bits to **measure** sound levels in a science investigation into the

that will allow a sprite to jump over obstacles doesn't work as the sprite does not jump high enough to clear the object. Learners can be challenged to **fix** the program to match the task description.

Suggested Activities

Learners create a piece of narrative writing on Scratch

Learners create a Space themed game on Scratch

Learners create a Clicker game using conditional repetition

Suggested Resources
Watch the assembly and then **explore** the micro:bit activities

Activity 1
Find out how technology can help save the planet

Tech for helping the planet

Then complete the follow up micro:bit tasks:

Max-min thermometer
Track highest and lowest temperatures by leaving this program running on a micro:bit.

Energy Light Timer
Time how long your lights are left on to **track** your energy use. You could also use this project to track hours of sunshine in a weather station project.

Which way now?
Shake the micro:bit and be given a random direction to walk. You will learn about variables, and using random numbers, selection and comparison logic blocks.

Activity 2
Find out about careers in technology

Tech for Entertainment, History and Sport

Then complete the follow up micro:bit tasks:

Counter
Create a simple project to help you **count**... skips, jumps, birds you see out of your window - anything!

Step Counter
Turn the BBC micro:bit into a step counter (or pedometer) to help you track how active you are - and learn some coding at the same time!

Low Energy Step Counter
Make a step counter with longer-lasting batteries.

Sensitive Step Counter
sound insulation properties of different materials.

Activity Picker
Finding it hard to decide or agree on what to do? Let this micro:bit program choose for you!

Times Table Tester
Test your knowledge of times tables with this project.

Activity 2
Explore how tech can be **used** for security with this activity before exploring cybersecurity in more depth.

Tech for Cybersecurity

Then complete the follow up micro:bit tasks:

Simple Door Alarm
Make an alarm to alert you to sneaky snoopers!

Pressure Switch Alarm
Create a wireless intruder alarm that will warn you when someone steps on a home-made pressure sensor.

Activity 3
Find out how technology can help save the planet

Tech for helping the planet

- Energy Cost Calculator
  Learn how to work out the cost of energy and make a timer that measures how much electric lights cost to run.

Activity 2
Find out about careers in gaming

Tech for Gaming and eSports

Then complete the follow up micro:bit tasks:

Group Teleporting Duck
Use the radio function to **create** a game of hot potato!

Make a micro:bit virtual pet
Code your own electronic pet and customise it to make it your own. The new micro:bit's built-in speaker makes it even more fun with the new expressive sounds.

Reaction Game
Make a reaction game with real physical switches you can bash as hard as you like!

Activity 3
Find out about careers in technology

Tech for Entertainment, History and Sport

Then complete the follow up micro:bit tasks:

Treasure Hunt
Use several micro:bits to make a physical treasure hunt game using radio communication.

TCH 2-13a TCH 2-14a TCH 2-14b TCH 2-15a

Create a step-counter you can make more accurate by tailoring it to your own walking style.

Guitar 1 Touch Tunes
Make touch buttons using tin foil and cardboard. Shape them to look like a keyboard or a guitar.

Guitar 2 Chords
Make the micro:bit guitar or keyboard play chords with a single touch.

Activity 3
Find out about how technology supports people to help others

Tech to Help People

Then complete the follow up micro:bit tasks:

Conductivity Tester
Use the micro:bit with two crocodile clip leads to investigate if a material conducts electricity - a simple way to carry out a classic science experiment!

Clap lights
Turn the micro:bit into a light that you can turn on and off by clapping or making any loud sound.

Light Alarm

Then complete the follow up micro:bit tasks:

Indoor-outdoor thermometer
Use two micro:bits to monitor indoor and outdoor temperatures.

Species Counter
Use the BBC micro:bit to help you count two different species of plants or animals in your school playground, garden or local park. You will learn about variables, and using the micro:bit's buttons and LED display.

Energy Light Meter
This light meter will help you **measure** how light levels vary around you when lights are turned on and off.

Activity 4
Find out about careers in technology

Tech for Entertainment, History and Sport

Then complete the follow up micro:bit tasks:

Fireflies
Turn a set of micro:bits into a magical glowing swarm of fireflies using radio communication

Other micro:bit activities:

Heartbeat beacon

Sound Logger
Make a sound level logger to monitor how loud or quiet different places around you get over time

Jukebox with Volume
This project adds a volume control to a micro:bit jukebox. It **plays** different tunes if you press button A or B, and you can also **adjust** the volume by **tilting** the micro:bit left or right.

Touch Timer
Make a simple timer using the new micro:bit's touch logo sensor.

Touch Stopwatch
Make a real stopwatch using the new micro:bit's touch logo sensor as an extra button.

Clap-o-meter
Measure how long applause - or any loud sound - lasts with this timer that uses the microphone on the new micro:bit.

Activity Array
Finding hard to decide or agree on what to do? **Use** arrays to **create** a micro:bit program that chooses for you!

Candle
Make an electronic candle that you can blow out! The new micro:bit's built-in microphone **detects** the sound of your breath and turns the candle off - and on again.

Sound Compass

TCH 2-13a TCH 2-14a TCH 2-14b TCH 2-15a

A radio-controlled remote alarm so you know when someone's turned the lights on or opened a drawer. *Other micro:bit activities:* Meet your micro:bit **Start** discovering some of the things the BBC micro:bit can do with this interactive investigation. • Individual lesson powerpoints for you to use to teach the concepts of Microbit	With two micro:bits you can keep track of a precious belonging or pet **using** radio messages. Teleporting Duck **Use** the radio function to make a duck fly invisibly through the air from one micro:bit to another. • Individual lesson powerpoints for you to use to teach the concepts of Microbit	**Make** a compass that makes a sound when you're pointing North to make it more accessible and useful. Proximity Beacon **Use** radio to sense how close another micro:bit is and then make a treasure hunt game. Guitar 3 Octaves **Improve** the micro:bit guitar by shifting the pitch up and down octaves. *Activity 4* **Explore** how tech can be **used** for security with this activity before exploring cybersecurity in more depth. Tech for Cybersecurity Then complete the follow up micro:bit task: Radio Door Alarm A wireless alarm to warn you when someone opens a door – or leaves it open. *Activity 5* Find out about how technology supports people to help others Tech to Help People Then complete the follow up micro:bit task: Passive Infrared Movement Alarm A wireless intruder alarm using a movement detector such as a burglar alarm or automatic lights in an office.

TCH 2-13a TCH 2-14a TCH 2-14b TCH 2-15a

Key Vocabulary	Cross Curricular Links
variables, fixed repetition, conditional repetition, parallel programmes, remix, debugging, bug, HTML, algorithm, creating, evaluation, logic, tinkering, persevering, collaborating, abstraction, decomposition, internet services	• I can show my understanding of how the number line extends to include numbers less than zero and have investigated how these numbers occur and are used. **MNU 2-04a** • Through practical activities which include the use of technology, I have developed my understanding of the link between compass points and angles and can describe, follow and record directions, routes and journeys using appropriate vocabulary. **MNU 2-17a** • I can use my knowledge of the coordinate system to plot and describe the location of a point on a grid. **MNU 2-18a** • When I engage with others, I can respond in ways appropriate to my role, show that I value others' contributions and use these to build on thinking. **LIT 2-02a** • As I listen or watch, I can make notes, organise these under suitable headings and use these to understand ideas and information and create new texts, using my own words as appropriate. (Game Design) **LIT 2-05a**

TCH 2-13a TCH 2-14a TCH 2-14b TCH 2-15a

Computational Thinking concepts and approaches	Computer Science Concepts and Approaches	Additional/Extension activities	Teacher CLPL Materials
• Decomposition • Abstraction • Algorithms • Debugging • Persevering • Collaborating • Creating • Logic • Evaluation • Persevering • Tinkering **Barefoot concepts and approaches**	• Variables • Sequences • Programming • Repetition	• Creative Computing Curriculum • Scratch Tinkering • BBC Bitesize Computing Science and ICT • Scratch cards - How to guides.pdf • Individual lesson powerpoints for you to use to teach the concepts of Microbit • Introduction to Microbit Challenges.pptx • Introduction to Scratch Challenges.pptx	• Teach Computing Science A Guide for Primary and Early Years Practitioners • DigiLearn Scot Computing Science Blog • Scratch for Educators • Barefoot Computing • Get Started with BBC Micro:bit • Code.org Careers in Computing Science

TCH 2-13a TCH 2-14a TCH 2-14b TCH 2-15a