Hove Learning Federation
Computing Policy

Adopted by Learning & Teaching Committee on behalf of the Governing body:

Amended: January 2025

We are committed to safeguarding and ensuring the health, safety and well-being of all pupils in accordance with safeguarding procedures and guidance for staff outlined in the school's Health and Safety, Child Protection, Security and Safeguarding policies.

Hove Learning Federation Computing Policy

Contents:

1. Introduction

A high-quality computing education equips pupils to use computational thinking and creativity to understand and change the world. The core of computing is computer science, in which pupils are taught the principles of information and computation, how digital systems work, and how to put this knowledge to use through programming. Building on this knowledge and understanding, pupils are equipped to use information technology to create programs, systems and a range of content. Computing also ensures that pupils become digitally literate – able to use, and express themselves and develop their ideas through, information and communication technology – at an age appropriate level and as actively safe and conscientious participants in a digital world.

2. Aims

At the Hove Learning Federation we aim to:

- to use technology with purpose, enjoyment and confidence
- to develop their use of software to be able to create a range of digital content
- to inspire pupil curiosity, creativity and understanding of computing as part of everyday life
- to identify situations and opportunities where the use of computing is relevant
- to begin to develop the necessary skills to use a range of devices, applications and programs
- to apply their learning and creativity to solve problems
- to be responsible digital citizens who understand how to keep themselves and others safe online *(for more details of Online Safety please see 'Hove Learning Federation Online Safety Policy')*
- to build on and apply prior learning to new tasks
- to encourage children's ability to think critically, reflect, debate and evaluate the use of technology

3. Intent

Throughout their time in the Hove Learning Federation, children will enjoy carefully and thoughtfully planned units. In all year groups, computing plays an important cross-curricular role.

Our children will experience confidence and enjoyment in computing and will gradually improve and master essential skills, developing a broad base of knowledge and layered understanding by the end of Key Stage 2. Children will be able to self-assess their own progress and will be given opportunities to reflect on their learning and discuss their understanding.

Subject teams will monitor the teaching and learning within computing carefully, ensuring children are making good progress and our curriculum is enjoyable, skills based and supports them to navigate the modern digital world.

4. Implementation

Each lesson contains revision, analysis and problem-solving. Through the sequence of lessons, we intend to inspire pupils to develop a love of the digital world, see its place in their future. Cross-curricular links are also important in supporting other areas of learning. Our lesson plans and resources help children to build on prior knowledge at the same time as introducing new skills and challenges. In EYFS, the focus is on learning basic computing skills and introducing digital literacy. In KS1, the focus is on developing the use of algorithms, programming and how technology can be used safely and purposefully. In KS2, lessons still focus on algorithms, programming and coding but in a more complex way and for different purposes. Children also develop their knowledge of computer networks, internet services and the safe and purposeful use of the internet and technology. Data Handling is featured more heavily in UKS2. Skills learnt through KS1 and LKS2 are used to support data presentation. Subject specific vocabulary is identified within each unit of work to support pupil understanding and explanations. Each unit of work is planned in a logical progression to build upon prior learning and apply existing knowledge and skills.

Across the Federation, each class is allocated time in computer suites. These times are used for the teaching of the computing units of work.

In the EYFS half of each class is taught in the suite at a time, with the other pupils remaining in the classroom with the classroom assistant. The two halves of the class switch over mid-session. Each pupil in EYFS works independently on a computer to ensure they are having ample practice to build up key computing skills such as mouse skills and logging on.

In KS1 and 2 the whole class accesses the computer suite at the same time. Pupils work in pairs and these pairs are selected by the adult to encourage cooperation, communication and collaboration between the partners whilst also allowing the pupils to support each other with their learning in computing.

As well as the school computer suites there is access to further technology. In EYFS and KS1, each class has a set of four iPads that can be used in the classroom to support cross curricular learning. In KS1 there is also a class set of Chromebooks that are currently being implemented to use within the Yr2 classes.

In KS2 there are several sets of Chromebooks and these are used across the Key stage to support cross curricular learning. Each pupil in Key Stage 2 has access to Google Classroom and has their own personal login which can be used to access learning both within the school environment and whilst at home. Teachers manage these Google Classrooms to share work and activities. Home Learning from Year 3 is set on Google Classroom but paper copies are also available for children if these are required.

5. Content

Aims and key principles:

Content and coverage for computing is plotted out in the Progression Ladder which shows the development in skills, knowledge and vocabulary from YR to Y6, and plots out the federation's spiral curriculum for this subject. This makes it clear to see where children have opportunities to revisit skills and understanding in order to achieve mastery.

Early Years and Key Stage 1

Teachers use the Computing Curriculum Map (*Appendix 2*) and Progression Ladder (*Appendix 5*), along with termly overviews and weekly plans, to set out the structure and content of computing lessons. Learning objectives are taken from the Early Years Foundation Stage and the National Curriculum and each lesson is carefully sequenced to build knowledge over time and are supplemented by the National Centre for Computing Education scheme of work.

Key Stage 2

As in Key Stage 1, teachers use the History Progression Ladder, along with termly overviews and weekly plans, to create lessons that cover the learning outlined in the National Curriculum. The knowledge and information that children are given in computing lessons has been carefully thought through by our computing subject team which are presented onto slides and shared in our Google classrooms and are supplemented by the National Centre for Computing Education scheme of work.

Expectations:	Strategies:
Learning objectives and statements related to the early learning goals are taught in the early years.	The Reception Year classes follow the Early Years Foundation Stage (EYFS) curriculum guidance, as well as 'Development Matters.' The revision of the EYFS in 2020 removed the 'technology' aspects from the framework. As a federation, we have agreed to continue to teach the skills and knowledge that young children need to support their progression into KS1, their use of technology in everyday life and the way in which technology can support learning in other EYFS areas. Our units of work in EYFS focus on mouse skills, technology to help us learn and digital literacy.

The National Curriculum objectives are used to support planning in KS1 and KS2.	• Core knowledge, skills and vocabulary are mapped in the progression ladder to ensure children are covering the correct content and knowledge is sequential (*Appendix 5*). • In Key Stage 1, the scheme of work has been developed from the Primary National Curriculum Programmes of Study for Computing, alongside our school's curriculum drivers: Standards, Engagement, Enquiry and Diversity - SEED (*Appendix 1*) and is supplemented by the National Centre for Computing Education scheme of work. Our computing objectives have been chosen to best suit the termly topics, and to reflect continuity and progression in children's knowledge, skills and understanding. Each unit focuses on a different strand within computing. Within each year group, pupils will be taught elements from all four strands of computing. o Programming o Online safety o Digital literacy (the use of software to create content) o Handling data
A clear spiral curriculum that progresses throughout each key stage builds on prior learning and introduces age-appropriate concepts, knowledge and skills.	• Please see (*Appendices 2 to 5*) for detailed progression ladder and curriculum content for EYFS, Key Stage 1 and Key Stage 2. • Pupils have opportunities to revisit skills and understanding in order to achieve mastery.
Computing links to other areas of the curriculum	Computing is a subject that links many subject areas together. For instance, internet searches are a fundamental part of research in most subjects, while skills in using authoring, spreadsheet and presentation software programs may be used to completing schoolwork and home learning. This becomes even more prevalent when pupils progress to KS3. Projects can be linked to other areas of the curriculum, perhaps using Humanities or Science themes, or a project-based approach. At Hove Learning Federation, we try to promote an integrated, cross-curricular approach, where computing content is embedded in other areas of the curriculum. A few clear examples are: the links between algorithms and maths, looping sequences and music, data capture and maths/science, as well as online safety and PSHE. Pupils in KS2 use Chromebooks in the classroom to support with maths and English interventions (Word Shark, Spelling Shed and Times Tables Rockstars). At Hove Learning Federation, we believe pupils should be given opportunities to apply and develop their computing capability by using information and communication technology (ICT) tools to support their learning in all subjects. Other examples of cross curricular computing skills include: <u>English</u> • Digital literacy promotes the skills of reading, writing, speaking and listening. • Pupils are encouraged to ask and answer their own questions • Pupils demonstrate their knowledge and understanding in a variety of ways including: digital posters, PowerPoint presentations and presented research • Pupils develop their understanding of a range of core vocabulary <u>Maths</u> • Data handling units • Programming (involving aspects of shape, space, position and direction) <u>PSHE and Citizenship</u> • Online safety • Digital Citizenship

	• Paired working <u>Design and Technology and Science</u> • SketchUp architectural design • Research based tasks <u>Creative and Art</u> • Graphics programs and Graphic design of digital products • Digital photography and videography • Generating art using Scratch and Inkscape
Computing lessons are hands on and active.	• Lessons are planned so that pupils have access to technology on a regular basis • Pupils are taught the skills to allow them to progress in their use of software and hardware across their time in school • Opportunities to apply skills and explore software are planned for • Pupils are actively encouraged to use problem solving skills to deal with technical issues, support their natural curiosity and develop new ways of tackling tasks
Year Group Specific Content	• The EYFS computing curriculum introduces young learners to fundamental digital skills through hands-on exploration. Children develop mouse skills to navigate digital environments and create images using basic drawing tools. They engage with technology to support learning across different areas, fostering curiosity and confidence. Early programming concepts are introduced through control activities and directional programming, helping children understand cause-and-effect relationships. Additionally, they explore digital media, using devices to capture and interact with multimedia content. These activities build foundational digital literacy, problem-solving, and creative expression, preparing children for further technological learning. • The Year 1 computing curriculum introduces students to fundamental digital skills, helping them recognize, use, and control technology responsibly. They explore digital tools through activities such as digital painting, writing, and data grouping, comparing digital and traditional methods. Students develop problem-solving skills by writing simple algorithms for floor robots and programming animations to enhance storytelling. These experiences build confidence in using technology, encourage creativity, and lay the groundwork for computational thinking and responsible digital practices. • The Year 2 computing curriculum focuses on identifying information technology in the world and understanding its responsible use. Students develop computational thinking by creating and debugging programs, using logical reasoning to predict outcomes. They explore digital music by composing rhythms and melodies on a computer. Data handling skills are introduced through collecting and organizing information into pictograms. Students also gain experience in digital photography, capturing and editing images for different purposes. Additionally, they design interactive quizzes, using events to trigger sequences of code. These activities enhance digital literacy, creativity, and problem-solving skills, preparing students for further technological learning. • The Year 3 computing curriculum expands students' digital skills by introducing them to stop-frame animation, where they capture and edit digital images to create animated stories. They develop programming skills by sequencing sounds in a block-based language and writing algorithms that use events to trigger actions. Data handling is further explored through branching databases, helping students classify objects using yes/no questions. Desktop publishing teaches them to create documents by modifying text, images, and layouts for specific purposes. Additionally, students learn about digital device connectivity,

	understanding how inputs, processes, and outputs work together to form networks. These activities enhance problem-solving, creativity, and digital literacy, preparing students for more advanced computing concepts. • The Year 4 computing curriculum deepens students' understanding of digital technology by exploring the internet as a network of networks, including the World Wide Web, and evaluating online content. Students develop audio production skills by capturing and editing audio to create podcasts while considering copyright. They enhance their programming knowledge through text-based coding, using count-controlled loops to draw shapes, and block-based coding to implement repetition in games. Data logging is introduced, enabling students to recognize why and how data is collected over time through investigations with data loggers. Additionally, they refine their digital creativity by manipulating and editing images, reflecting on their impact and intended purpose. These activities foster critical thinking, creativity, and problem-solving skills, supporting their progression in computing. • The Year 5 computing curriculum enhances students' understanding of IT systems and their role in enabling internet searches. They develop data management skills by using flat-file databases to organise and analyze data, creating charts to answer questions. Video production introduces planning, capturing, and editing techniques to produce short films. Students explore vector graphics, using layers and grouped objects to create digital images. Physical computing is introduced through programmable microcontrollers, where students explore conditions and selection. Additionally, they apply selection in programming by designing and coding interactive quizzes. These activities strengthen logical thinking, digital creativity, and problem-solving skills, preparing students for advanced computing concepts.

	• The Year 6 computing curriculum focuses on advanced digital skills, fostering collaboration and problem-solving. Students explore how data is transferred by working collaboratively online. They develop web design skills, creating webpages while considering copyright, aesthetics, and navigation. Programming knowledge is expanded through variables in game design, enhancing interactivity. Spreadsheet use is introduced, enabling students to organize and analyse data effectively. They engage in 3D modelling, planning and developing digital representations of physical objects. Additionally, students explore physical computing by designing and coding projects that capture inputs from physical devices. These activities refine critical thinking, creativity, and technical proficiency, preparing students for future digital challenges. • *For further details regarding Online Safety and Hove Learning Federation's procedures regarding this, please see 'Hove Learning Federation Online Safety Policy'*

6. Assessment

Aims and key principles:

Monitoring children's attainment, understanding and acquisition of skills in our foundation subjects is essential to ensuring we can provide children with the support and challenge they need to access, and flourish within, our curriculum. In history, assessment is carried out using a variety of methods as outlined below. Each teacher will be aware of monitoring this through other subjects where an understanding of history and the ability to use the skills of a historian can be employed.

Expectations:	Strategies:
Assessment is linked to planning and is used to inform future provision, teaching and learning	• Teachers plan lessons linked to National Curriculum Objectives. • Previous learning is revisited at the start of every lesson as a form of assessment and to support long term memory retention. • Subject leaders monitor planning and assessment across key stages to ensure knowledge and skills are mapped out across year groups.

	• Knowledge and skills are built on to ensure any gaps are addressed. • Learning objectives and targets for each lesson are shared with the children and assessed at the end of each lesson and unit.
Formative assessment is continually on going to support our understanding of children's progress	• In Early Years, teachers are continuously observing children's independent learning in the learning environment. This includes working independently and collaboratively and is recorded to provide the evidence that informs teacher assessment. • As children progress through Key Stage 1, they will learn to self-assess their work against the objectives for that lesson/unit of work alongside the teacher assessment. This allows children to take ownership of their learning and ensures they understand the lesson objective clearly. • In Key Stage 2 children's understanding is monitored through Assessment for Learning, pupil voice and teacher observation, and support is carefully planned in where appropriate to ensure learning is inclusive.
Assessment is evidenced in books in KS1 and KS2	At Hove Learning Federation, we support our children by assessing in different ways as they move through the year groups and progress through the curriculum. Our assessments are used as tools to help us assess skills and knowledge, target learning, provide challenge and deepen understanding. • During KS1, where the children are learning to structure their work and are developing the ability to reflect and assess their achievements within each lesson, we have a bespoke sheet for each session. These activity sheets include the main task and challenge opportunity along with three differentiated learning outcomes that the children can self-assess against. The teacher will then also tick these. • As the pupils move into KS2 and focus on a self-awareness of their developing knowledge and understanding, we begin each unit by exploring the relevant Core Knowledge file. This may contain knowledge from units in previous years where the current unit builds upon their use of a specific application as well as the new knowledge for the unit. The Core Knowledge files also contain key skills related to the unit. At the end of each lesson or unit, pupils self-assess their confidence with regard to the key skills by circling a relevant symbol (see Appendix 6).
Assessment and monitoring are ongoing to support future planning	• In the Early Years Foundation Stage, the children's knowledge, skills and understanding will be assessed using 'Development Matters,' and the Early Years Learning Goals (predominantly within 'Understanding the World'). These are recorded half termly into Target Tracker and then this information is used towards the end of year feedback given to parents and the final EYFS Profile. • At Key Stage 1, children's progress and attainment is tracked against age-related expectations. Assessment boxes are included in books and teachers assess children based on learning objectives. The learning objectives for each lesson provide a clear focus for assessment. At the end of each school year, every child will be assessed and recorded on Arbour, as: 'working below', 'working towards', 'secure' or 'secure plus'. These levels are communicated to parents in the end of year school report. At the end of Year 1, the assessments will be passed to the Year 2 teacher. At the end of Year 2, the accrued assessments will be used by the teacher to make a judgement about each child's ability in computing across the key stage. • At Key Stage 2, teachers assess children's understanding of computing and the knowledge they have acquired within each unit. The learning objectives for each lesson provide a clear focus for the assessment of

	their skills. Assessment for Learning strategies are utilised during lessons and to review pupil outcomes. Teachers also review pupil self-assessments to inform subsequent lessons and future planning.

7. Skills

Aims and key principles:

The teaching of computing skills is carefully mapped across the key stages. The children are gradually introduced to the skills alongside the acquisition of substantive knowledge. As computing knowledge and vocabulary are developed through our spiral curriculum, the children have the opportunity to revisit, embed and build on understanding.

Expectations:	Strategies:
Computing skills are embedded in each lesson and unit and carefully mapped out across the school	• Computing skills are taught alongside the knowledge acquisition aspect of the lesson. • Activity sheets in Key Stage 1 include a section for children to assess the skills they have used in each session. • In Key stage 2, learning objectives are based on computing skills and children are given opportunities to discuss and assess their confidence with the acquisition of these skills. • Activities are planned to ensure children are able to use computing skills to progress their knowledge • Pictures and quotes of children in dedicated computing lessons and using devices in class are included in books and on the EYFS portal.
Computing skills progress across the Key Stages	• Progression in computing skills is included in the Progression Ladder. • Children are aware of the skills they have been using in computing across the key stages and are encouraged to discuss how they have supported their growing understanding. • Evidence in books shows the variety of opportunities for using and applying these skills across both Key Stage 1 and 2 and The Portal.
Children with SEN or with EAL know the variety of working historically skills they are using to develop their knowledge and understanding with support	• Appropriate support is given to children with SEN and EAL children so they can access the learning and understand the skills needed to complete tasks. • Dual coding (using symbols from the Noun Project) is used to support the acquisition of specific vocabulary and is currently included on lesson slides and core knowledge files in Key Stage 2. • Higher level challenge partners are used to ensure children with SEN and or EAL are provided with high quality talk and modelled language of computing skills. • Activities ensure children with SEN or EAL can access tasks appropriately and share their understanding of computing concepts. • Differentiation and scaffolds are included where appropriate to enable access to learning and ensure children make at least expected progress. Pictures and quotes are taken from children with SEN and or EAL to ensure evidence is recorded in books and on The Portal (EYFS).

8. Knowledge and Vocabulary

Aims and key principles:

Our teaching of computing knowledge and vocabulary is carefully mapped to ensure it is delivered in a manner which will reduce cognitive overload and maximise children's understanding and retention. Knowledge and vocabulary acquisition builds gradually and in a spiral approach, which deepens understanding and encourages individual reflection and exploration of ideas.

Expectations:	Strategies:
Computing vocabulary linked to each unit is included in the Computing Progression Ladder, Core Knowledge Files and Curriculum Maps and progresses with the associated knowledge	• Children are taught the specific computing vocabulary as prescribed in the National Curriculum and Development Matters. • Vocabulary is dual coded, to support understanding and recall. • Vocabulary is explored in a variety of ways, including etymological observations, encouraging links to be made within and between computing topics and other subjects. • Children will become confident in using computing terms, and will gradually broaden the range of vocabulary used in cross-curricular work. • In Key Stage 2, Core Knowledge Files may be shared with parents and carers at the start of a unit, to encourage wider discussion and greater familiarity.
Vocabulary is included in each session and progression across sessions and year groups is evident	• Vocabulary is unpicked and explicitly taught each lesson and can be seen on flips/slides. • Children are given time in each lesson to hear and say key vocabulary and question the understanding of key words. • Key vocabulary from prior learning is discussed in the connecting learning flip/slide to embed vocabulary in long term memory. • Flips/slides show clear progression of vocabulary across the key stages. • In Key Stage 1, vocabulary is included on activity sheets in books and high expectations ensure children use this vocabulary in explaining their knowledge and understanding.
Misconceptions in knowledge and vocabulary are picked up early and addressed within lessons or before the subsequent lesson.	• Teachers plan for misconceptions to ensure correct knowledge and vocabulary is taught and understood each lesson. • Adults are confident to pick up on misconceptions in knowledge and vocabulary that the children may have and ensure these are addressed early and clearly. • Higher order questions, challenges and visual prompts are used in lessons to assess and support misconceptions. • Grumpy Frog is used on flips in Key Stage 1 to pre-empt possible misconceptions and address these to support whole class knowledge and understanding.

9. Equal Opportunities, Inclusion and Access

At Hove Learning Federation, we use Quality First Teaching to consistently meet the needs of all pupils. This includes ongoing assessment for learning which guides the path of the lesson, adapting lessons and responding to the needs of the children accordingly.

Challenge activities are included throughout computing lessons to indicate to children how they can deepen their knowledge. In Early Years and Key Stage 1, school characters are used (such as 'scuba diver challenge', 'submarine challenge' and 'Professor Prove-It'), while Key Stage 2 use a range of challenges and higher order questioning.

New topic-specific computing vocabulary (tier 2 or tier 3) is taught using symbols and images (dual coding) to support children's understanding and are pre-taught to children before a lesson, where relevant and purposeful. This vocabulary is displayed on lesson slides and is revisited throughout a unit, and built on year upon year.

Where appropriate in computing lessons, children are provided with tasks that have been broken down into small steps, giving them achievable goals. Some children may be offered a choice of how to record their work in different

ways, such as with a digital camera/ verbally/ with a tape-recorder. At times, tasks are designed so that outcomes can be child-led allowing for a variety of responses and ownership of learning.

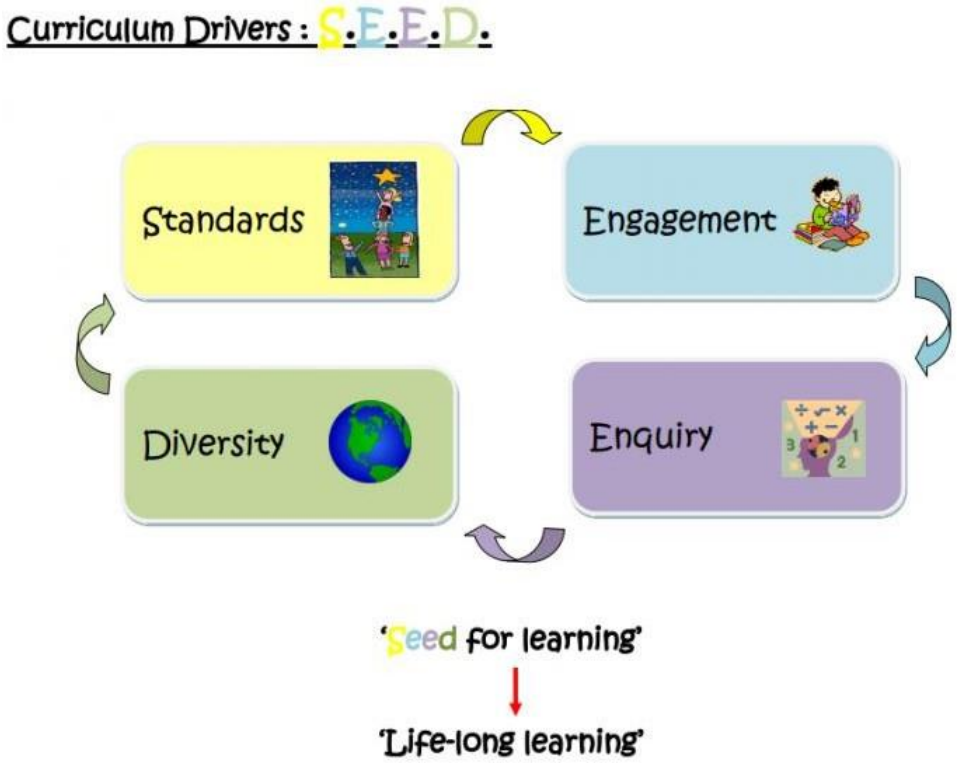
Additional materials can be provided to support learning (for example visual aids such as photographs, Makaton symbols, concept boards, dual coding, Communicate in Print resources, adapted scissors or other tools, or larger scale resources). Adult support is allocated to guide learning where needed and we ensure that all children can access trips.

Finally, mutual respect and tolerance for all cultures and different family units will be promoted through the study of computing. Our topics are designed and structured to celebrate different cultures and traditions. People from a range of different races, cultures and backgrounds and their contributions to the development of computing and communication technology will be learnt about. In this way, all children will be enabled to achieve their full potential.

10. Appendices

Appendix 1: Key Stage 1 Curriculum Drivers (SEED)

https://www.westhoveinfants.co.uk/our-curriculum/curriculum-drivers/



Appendix 2: curriculum map for KS1 and KS2

WEST HOVE INFANT SCHOOL
A family of friends

HOVE JUNIOR SCHOOL
LEARN

Computing Curriculum Sequence Overview 24/25

	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
Reception	**Mouse Skills**	**Creating Images**	**Learning Through Technology**	**Control**	**Programming and Direction**	**Digital Media**
Year 1	**Technology Around Us** Recognising technology in school and using it responsibly.	**Digital Painting** Choosing appropriate tools in a program to create art, and making comparisons with working non-digitally.	**Digital Writing** Using a computer to create and format text, before comparing to writing non-digitally.	**Grouping Data** Exploring object labels, then using them to sort and group objects by their properties.	**Moving a Robot** Writing short algorithms and programs for floor robots, and predicting program outcomes.	**Programming Animations** Designing and programming the movement of a character on screen to tell stories.
Year 2	**Information Technology Around Us** Identifying IT and how its responsible use improves our world in school and beyond.	**Robot Algorithms** Creating and debugging programs, and using logical reasoning to make predictions.	**Digital Music** Using a computer as a tool to explore rhythms and melodies, before creating a musical composition.	**Pictograms** Collecting data in tally charts and using attributes to organise and present data on a computer.	**Digital Photography** Capturing and changing digital photographs for different purposes	**Programming Quizzes** Designing algorithms and programs that use events to trigger sequences of code to make an interactive quiz.
Year 3	**Stop-frame Animation** Capturing and editing digital still images to produce a stop-frame animation that tells a story.	**Sequencing Sounds** Creating sequences in a block-based programming language to make music.	**Branching Databases** Building and using branching databases to group objects using yes/no questions.	**Desktop Publishing** Creating Documents by modifying text, images and page layouts for a specified purpose.	**Events and Actions in Programs** Writing algorithms and programs that use a range of events to trigger sequences of actions.	**Connecting Computers** Identifying that digital devices have inputs, processes, and outputs, and how devices can be connected to make networks.
Year 4	**The Internet** Recognising the internet as a network of networks including the WWW, and why should evaluate online content.	**Audio Production** Capturing and editing audio to produce a podcast, ensuing that copyright is considered.	**Repetition in Shapes** Using a text-based programming language to explore count-controlled loops when drawing shapes.	**Data Logging** Recognising how and why data is collected over time, before using data loggers to carry out an investigation.	**Photo Editing** Manipulating digital images, and reflecting on the impact of changes and whether the required purpose is fulfilled.	**Repetition in Games** Using a block-based programming language to explore count-controlled and infinite loops when creating a game.

Year 5	**System and Searching** Recognising IT systems in the world and how some can enable searching on the internet.	**Flat-file Databases** Using a database to order data and create charts to answer questions.	**Video Production** Planning, capturing and editing video to produce a short film.	**Introduction to Vector Graphics** Creating images in a drawing program by using layers and groups of objects.	**Selection in Physical Computing** Exploring conditions and selection using a programmable microcontroller.	**Selection in Quizzes** Exploring selection in programming to design and code an interactive quiz.
Year 6	**Communication and Collaboration** Exploring how data is transferred by working collaboratively online.	**Webpage Creation** Designing and creating webpages, giving consideration to copyright, aesthetics and navigation.	**Variables in Games** Exploring variables when designing and coding a game.	**Introduction to Spreadsheets** Answering questions by using spreadsheets to organise and calculate data.	**3D Modelling** Planning, developing and evaluating 3D computer models of physical objects.	**Sensing Movement** Designing and coding a project that captures inputs from a physical device.

Appendix 3: Example of Year 2 end of unit assessment sheetscratc

1

Name *

Enter your answer

2

What do we use a tally chart for? * (1 Point)

Animal	Tally	Total
	IIII	4
	𝍸 I	6
	𝍸 III	8
	III	3

- ◯ To draw pictures of animals
- ◯ To record and count data
- ◯ To help us write stories
- ◯ To make a shopping list

Appendix 4:Example of Hove Learning Federation Computing Progression Ladder

Plain text = Curriculum Expectations (please **DO NOT** change or delete these), *Italic = Additional WHIS/HJS*

	Year R (Computing)		Year 1	Year 2	Year 3	Year 4	Year 5	Year 6
	Nursery	Reception						
	By the end of Reception: There are Computing ELG or statements in the EYFS however the computing we teach in school is supporting children in other areas of the EYFS and providing skills to access an increasingly digital world.		**By the end of KS1:** In Year 1 and 2 computing, students develop foundational programming and problem-solving skills through sequencing, predicting outcomes, and debugging. They learn to match commands to results, give and follow clear instructions, and experiment with movement commands to control devices and robots. They explore algorithms by planning and testing sequences, recognizing patterns, and using different programming tools to achieve similar outcomes. Through designing projects, selecting sprites, backgrounds, and programming blocks, they create and refine digital designs while debugging and improving their work. These skills foster logical thinking, creativity, and an understanding of how computers follow instructions to complete tasks.		**By the end of Lower Key Stage 2:** In Year 3 and 4 computing, students expand their programming knowledge by exploring Scratch, text-based coding, and loops. They learn that objects in Scratch have attributes, recognize commands as blocks, and create programs by typing commands. They design and test algorithms, use count-controlled loops, and modify code to achieve specific outcomes. Through sequencing, repetition, and event-driven programming, they refine their ability to predict, debug, and improve programs. They make design choices, incorporate sound and movement, and evaluate the effectiveness of their code. By reusing and adapting existing code snippets, they develop logical thinking and problem-solving skills in coding projects.		**By the end of Upper Key Stage 2:** In Year 5 and 6 computing, students develop more advanced programming skills, including working with microcontrollers, variables, loops, and selection statements. They learn to create circuits, program LEDs, and control multiple outputs using loops. They explore variables as placeholders for data, modifying them through conditions and user input. By implementing selection statements (if...then...else), they control program flow and create interactive projects. They design, test, debug, and refine programs, considering real-world applications and improving efficiency. Through experimentation with different inputs, debugging strategies, and structured program design, they enhance their ability to create, evaluate, and share sophisticated coding projects.	
Programming A and B **Y1 Programming A: Moving a robot** **Y1 Programming B: Animations** **Y2 Programming A: Robot Algorithms** **Y2 Programming B: Programming quizzes**	*Copy the actions of others to operate simple equipment and toys*	*Help adults operate equipment around the school, independently operating simple equipment*	-I can match a command to an outcome - I can predict the outcome of a command on a device - I can run a command on a device	-I can choose a series of words that can be enacted as a sequence - I can follow instructions given by someone else - I can give clear instructions	-I can explain that objects in Scratch have attributes (linked to) - I can identify the objects in a Scratch project (sprites, backdrops) - I can recognise that commands in Scratch are represented as blocks	-I can create a code snippet for a given purpose - I can explain the effect of changing a value of a command - I can program a computer by typing commands	-I can create a simple circuit and connect it to a microcontroller - I can explain what an infinite loop does - I can program a microcontroller to make an LED switch on	-I can explain that the way a variable changes can be defined - I can identify examples of information that is variable - I can identify that variables can hold numbers or letters
	Explore simple software to make things happen	*Use simple software to make things happen*	-I can follow an instruction - I can give directions - I can recall words that can be acted out	-I can show the difference in outcomes between two sequences that consist of the same commands - I can use an algorithm to program a sequence on a floor robot - I can use the same instructions to create different algorithms	-I can choose a word which describes an on-screen action for my plan - I can create a program following a design - I can identify that each sprite is controlled by the commands I choose	-I can test my algorithm in a text-based language - I can use a template to create a design for my program - I can write an algorithm to produce a given outcome	-I can connect more than one output component to a microcontroller - I can design sequences that use count-controlled loops - I can use a count-controlled loop to control outputs	-I can explain that a variable has a name and a value - I can identify a program variable as a placeholder in memory for a single value - I can recognise that the value of a variable can be changed
	Use buttons on electronic toys and be able to state what the buttons do.	*Press buttons on a floor robot or screen robot and talk about the movements*	-I can compare forwards and backwards movements - I can predict the outcome of a sequence involving forwards and backwards commands - I can start a sequence from the same place	-I can compare my prediction to the program outcome - I can follow a sequence - I can predict the outcome of a sequence	-I can create a sequence of connected commands - I can explain that the objects in my project will respond exactly to the code - I can start a program in different ways	-I can identify everyday tasks that include repetition as part of a sequence, eg brushing teeth, dance moves - I can identify patterns in a sequence - I can use a count-controlled loop to produce a given outcome	-I can design a conditional loop - I can explain that a condition is either true or false - I can program a microcontroller to respond to an input	-I can decide where in a program to change a variable - I can make use of an event in a program to set a variable - I can recognise that the value of a variable can be used by a program
	Identify some differences between a variety of toys.	*Explore options and make choices with toys, software and websites*	-I can compare left and right turns - I can experiment with turn and move commands to move a robot - I can predict the outcome of a sequence involving up to four commands	-I can explain the choices I made for my mat design - I can identify different routes around my mat - I can test my mat to make sure that it is usable	-I can combine sound commands - I can explain what a sequence is - I can order notes into a sequence	-I can choose which values to change in a loop - I can identify the effect of changing the number of times a task is repeated - I can predict the outcome of a program containing a count-controlled loop	-I can explain that a condition being met can start an action - I can identify a condition and an action in my project - I can use selection (an 'if...then...' statement) to direct the flow of a program	-I can choose the artwork for my project - I can create algorithms for my project - I can explain my design choices
			-I can choose the order of commands in a sequence - I can debug my program - I can explain what my program should do	-I can create an algorithm to meet my goal - I can explain what my algorithm should achieve - I can use my algorithm to create a program	-I can build a sequence of commands - I can decide the actions for each sprite in a program - I can make design choices for my artwork	-I can explain that a computer can repeatedly call a procedure - I can identify 'chunks' of actions in the real world - I can use a procedure in a program	-I can create a detailed drawing of my project - I can describe what my project will do - I can identify a real-world example of a condition starting an action	-I can choose a name that identifies the role of a variable - I can create the artwork for my project - I can test the code that I have written