Holly Hill Primary School

Computing Policy

1. Subject Intent

General Statement

At Holly Hill we believe that a quality Computing curriculum should be broad and balanced across a range of subjects and equip all learners including those from disadvantaged groups to use computational thinking and creativity to understand and change the world.

The Computing curriculum at Holly Hill enables children to become effective, creative, and positive citizens in a digital world. Computer science is the foundation of the Computing curriculum, where the principles of how digital technology works will be taught, alongside the practical experience of programming. Information technology enables the children to apply their knowledge, skills, and approaches to purposefully create and make digital content. We aim to empower the children through digital literacy to become balanced, resilient, and responsible members of the community. This will allow them to be able to use and express themselves, develop their ideas, make informative decisions through information and communication technology.

Through our Computing curriculum we make links to our whole school curriculum intent statement. The aspects which are particularly significant to Computing are:

- Preparation for adult life and work
- Balance of knowledge and skills
- Emotional well-being and healthy relationships
- Experiences and opportunities

Specific Aims

The National Curriculum outlines the following aims for Computing:

- Can understand and apply the fundamental principles and concepts of computer science, including abstraction, logic, algorithms, and data representation.
- Can analyse problems in computational terms, and have repeated practical experience of writing computer programs in order to solve such problems.
- Can evaluate and apply information technology, including new or unfamiliar technologies, analytically to solve problems.
- Are responsible, competent, confident, and creative users of information and communication technology.

At Holly Hill, the Computing curriculum is designed so that we meet the National Curriculum aims through a progression of skills and knowledge taught in the sequence below:

	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
EYFS	Computational thinking within focused sessions and continuous provision Use of laptops (logging on and accessing age-appropriate websites), iPads, Bee-Bots, Cameras and Remote-control devices					
KS1	Cycle A Enchanted World – focused	Cycle A Toy Story - focused	Cycle B How does it work - focused	Cycle B Mother Nature - focused	Cycle A Who lives in a place like this? – enhancement music link	Cycle A Oh I do like to be beside the seaside - enhancement Cycle B London Rocks - enhancement
LKS2	Cycle B Superheroes - focused	Cycle A Festival of Light - focused Cycle B We are family enhancement	Cycle A What's wonderful about Stevie? focused Cycle B Bizarre Bodies - enhancement		Cycle A Stone Age - enhancement Cycle B Dungeons and Dragons - focused	
UKS2	Cycle A Great Explorers - enhancement DT / focused Computer aided design / 3D modelling and enhancement Cycle B Clockwork - -focused	Cycle A Crime & Punishment - enhancement	Cycle A How to save the human race -focused Cycle B The Golden Age - enhancement			Cycle B Double, Double, Toil &Trouble – focused

Computing Curriculum Design

COMPUTING	CYCLE A		CYCLE B	
YEAR 1/2	Programming	Creating Media - Painting & writing	Programming	Technology around us
YEAR 3/4	Programming	Creating Media - Audio production	Programming	Systems and networks
YEAR 5/6	Programming	Creating Media - 3D modelling	Programming	Systems and networks

Rationale

At Holly Hill, Computing coverage is organised in this way because the Computing curriculum can be split into 3 broad areas: - Computer Science, Information Technology and Digital literacy. We wish to cover the Computing curriculum *over 4 main themes: -*

Computer systems and networks, programming (incl. algorithms), data and information and creating media. These are bound together with safety and security, effective use of tools and the impact of technology. We focus on these 4 main themes to group the curriculum based on the skills and concepts to be covered across the Computing Curriculum. The coverage is balanced between focused units and using Computing as an enhancement to other curriculum areas. Our coverage involves concepts and skills that are taught in a specific order to build on prior learning and experiences over the two-year planning cycle. Other experiences and opportunities will also be available to support all our learners. Online safety is taught both through Computing sessions (Education for a Connected World, via Project Evolve, Google Legends) and is also part of our Jigsaw PHSE scheme of work. We also take part in Safer Internet Day on a yearly basis to promote online safety across the school.

Computational thinking concepts and approaches are introduced within our Early Years Foundation Stage and continued through both Key Stages to support the fundamental skills of problem solving that links to both everyday life skills and computer science. Within the Computing curriculum Computer Science (algorithms and programming) have been allocated more time to ensure the specific skills and concepts can be covered across the two-year cycle.

2. Implementation

Teaching and Learning

Because at Holly Hill Primary and Nursery we understand that learning takes place when there is a change to long-term memory, we outline the key knowledge for a unit of work on a knowledge organiser. All of our knowledge can be found on our Computing sticky knowledge map. In Computing, key knowledge is knowledge that relates directly to the National Curriculum attainment targets. Progression of skills is mapped out using our progress maps.

Computing will be taught by a qualified teacher or computing leads who have planned, resourced, and fully understand how to deliver the lesson, to cover everybody's needs and abilities. Support is available from the Computing subject leads regarding planning, assessment etc.
Computing is to be taught both discretely and thematically via cross-curricular links over the two-year planning cycle. Computing concepts and skills such as algorithms and programming, computing systems and networks will be taught through specific units, over a two-year cycle to enable deeper knowledge and

understanding to be taught. This will ensure computer science skills and concepts are both progressive and build from prior learning and experiences within each block of year groups (KS1, LKS2, UKS2). Other areas of computing will be taught cross-curricular with explicit links to the computing objectives to ensure coverage across the curriculum, and to enable the students to see the links between other curriculum areas and the use of technology.
The frequency and length of each lesson will vary dependant on the theme, skills or concepts being taught.

The Teach Computing Curriculum (created by the Raspberry Pi Foundation on behalf of the National Centre for Computing Education - NCCE) will be the solid base on which our Computing curriculum will be centred. This base then allows Holly Hill Primary School to be creative and tailor the resources to meet our school community.

An effective teaching sequence will involve the pupils having conceptual understanding of exploring what they are doing as well as when and why. This will be taught alongside developing the skills (how) to accomplish it.

The essential element of every Computing lesson will start with an Online Safety knowledge check to ensure that the children have a secure understanding of how to stay safe online and are aware of expectations when using school equipment online.

Our pedagogy will include leading with concepts by introducing concepts, vocabulary, or terms of knowledge before exploring how they are applied and these build upon previous lessons or previous phase of lessons.

The sequence may be introduced through an abstract activity then moving onto concrete or real-life experiences, or relating to another subject or unplugged activity. Analogies support the comparing of one thing to another, for example: - pupils passing messages and creating a web using string to represent computer system networks.

When teaching algorithms and programming using the PRIMM (Predict, Run, Investigate, Modify, Make) approach supports reading and investigating sequences of code in order to modify and make programs.

Another method includes using ‘levels of abstraction - LOA,’ to consider specific skills or concepts without accidentally adding more detail.

The levels are Task – what is needed, Design – what should it do, Code – how it is done, run (running the code) – what it does.

Other pedagogical content knowledge methods used within computing will include

1. Lead with concepts
2. Unpack, unplug, repack
3. Create projects
4. Challenge misconceptions
5. Structure lessons
6. Work together
7. Model everything
8. Add variety
9. Make concrete
10. Read and explore code first
11. Get hands on
12. Foster program comprehension

Vocabulary and computing terms will be introduced and discussed both within the computing lesson and during cross curricular links.

Lessons will feature formative methods of assessment for example strategic questioning, self or peer assessment, pupil feedback or mini plenaries.

All classes will have access to a class set of laptops, shared resources of iPads, Bee-Bots, Probots, Micro:Bits Remote Control devices (FS), Cameras (F2) and a Data Logger / Apps. Additional Micro:Bits can also be loaned from our Computing Hub.

Every classroom will display a poster showing computational thinking concepts and approaches and F2-Yr 4 will display the pupil acceptable use policy as agreed to by the class. Children in Year 5 and 6 will sign an individual copy of the UPKS2 Acceptable Use Policy, which will be stuck into their creative arts book.

SEND

Also see document on Computer Science by Barefoot Computing, 'Computing for SEND-Guidance for Teachers.' Support is also available from the SENCO / SEND (Special Educational Needs and Disabilities) Team or the Computing leads.

Sensory, Physical and Medical

In Computing barriers could include hypo/hypersensitivity to noise or other sensory stimuli e.g. visuals from the computer/laptop screens, in built sounds from programs and online teaching platforms or the physical environment. Other barriers could include visual impairments, hearing impairments, motor difficulties such as a loss of a limb or having a medical condition such as epilepsy.

To support children with Sensory, Physical and Medical needs in Computing, strategies could include using ear defenders or headphones, use of a sans serif font e.g. comic sans or Verdana, use of an increased size of text on the screen or interactive white board, subtitles or voice overs, modified accessories such as a lower case keyboard, or large keys keyboard or keyguard or mouse/rollerball mouse, change of settings e.g. brightness of the screen, back ground colours or high contrast between text and background and live captions. Use of unplugged activities with familiar contact and sensory approaches.

Cognition and Learning

In Computing barriers could include difficulties accessing text-based resources or instructions or presentations, using standard keyboards may slow down the output due to difficulties with uppercase/lowercase recognition and that certain login details screens timeout before the text may have been entered. Working memory difficulties, and difficulties following a sequence of instructions.

To support children with Cognition and Learning needs in Computing, strategies could include using symbolised guides or task ladders to differentiate tasks, extra time to process and be aware of cognitive load when introducing new concepts. Introduce new content in smaller chunks, present new information using familiar concepts where possible and practice what has been learnt before moving on. Teach key skills explicitly in advance to aid recall. Review previous learning at regular intervals. Provide planning templates and avoid 'blank canvas' tasks. Help pupils to remember and recall language through physical actions, songs, signs or symbols, word banks, key vocabulary pre-teach. Lower-case keyboards, use of Clicker 7 software. Use assistive technology or apps that are inbuilt to devices or programs e.g. magnifier or zoom settings, predictive text, speech to text (dictate) or text to speech (read aloud) supporting new vocabulary or concepts with symbols or supporting images. Immersive reader can be used within OneNote to support the access of text.

Provide opportunities for the pupils to express their learning through multimedia e.g. video, presentations rather than text-based reports. Allow time to tinker with completed programs to see what they do before they modify code. Pair programming, PRIMM, UMC and Parsons problems support coding activities.

Social, Emotional and Mental Health

In Computing barriers could include an overload of information, difficulties with concentration.

To support children with Social, Emotional and Mental Health needs in Computing strategies, could include movement breaks, physical unplugged activities, expression of ideas through multimedia, use of task ladders to break down tasks into smaller steps and visual prompts being available to support with more complex tasks, such as scratch programming. To support and build resilience ensure pupils achieve an outcome quickly and then build on it. Celebrate failure as part of the learning process through debugging/decomposition. Pair programming, PRIMM, Use-Modify-Create (UMC) and Parson's problems support coding activities.

Communication and Interaction

In Computing barriers could include the understanding, expressing, or recalling of vocabulary of computing terminology, especially regarding computer science. Other barriers could include the sharing of resources, the completion of task in a specific way or general organisational difficulties. Difficulties with the generalising of knowledge and transferring skills between programs or tasks could also be a barrier.

To support children with Communication and Interaction needs in Computing, strategies could include using visual supports such as symbolised guides or photographs, timers (digital or physical), task ladders to differentiate tasks, now and next boards, pre-teach/introduce beforehand of new software/programmes. Providing an overview of the project or outcome, before breaking it down using a planning sheet e.g., we are going to make a movie about plants, first we are going to take the photographs. Use flow diagrams, mind maps, or act out algorithms. Allow time to tinker with completed programs to see what they do before they modify code. Provide clear models and complete examples.

Teach key skills explicitly in advance to aid recall. Use technology that has different outputs e.g. Bee-Bot / Pro-Bot (movement), Scratch piano (music). Help pupils to remember and recall language through physical actions, songs, signs, or symbols. Use unplugged activities with familiar contact and sensory approaches.

Good practice for SEND is good practice for all pupils: -

Provide clear outcomes and expectations

Pair programming

Targeted questioning

Provide feedback

Formative assessment

Opportunities to apply specific knowledge in multiple ways

Encourage independent learning

Celebrate success and share pupils' work

Make it fun and relevant (engage individual pupils with the learning)

Experiences and Opportunities in Computing

In Computing during the school year, we offer: -

After School Clubs

Focus days eg Safer Internet Day, No Pens Wednesday

Joining live stream events eg Google Legends, British Science Week, Code Club live coding

3. Impact

Assessment

In Computing we ensure progression through the use of knowledge organisers, progress maps, NCCE Teach Computing, Project Evolve and National Curriculum Expectations.

In Computing we assess the children formatively with key knowledge reviews and quizzes, observations, and the implementation of the feedback policy. Each lesson in a sequence begins with a 'knowledge check-up,' which provides opportunities for pupils to recall the key knowledge from the previous lessons in that unit.

In Computing we assess the children at the end of each unit. Assessment judgements are based on the evidence of knowledge in the child's 'end point.' In Computing an end point could be a final product e.g., scratch project, video, presentation (digitally or orally), explanatory conversation or a quiz.

Staff will use their assessment judgements to complete an evaluation of the learning in that unit, which is submitted to the subject leader. Where gaps or issues have been identified these will be acted upon through adaptation of later units, adaptation of planning, further knowledge reviews or the introduction of an afterschool club.

Monitoring and Evaluation

- Evidence of work can be found in a child's shared subject book, Learning Journal/Tapestry (EYFS) which they keep for two years across the key stage or within their personal digital folder (Digital folder – across whole school life).
- Subject leaders will use a combination of evidence from book looks, learning walks, environment checks / displays and end of unit data to monitor the standards in their subject and inform the action plans for curriculum development.
- Subject leaders will use the unit evaluations to inform their action plans, evaluate learning from the Knowledge organisers or identify if any CPD is wanted/needed.

What does the impact of Computing look like at Holly Hill?

Based on intent, children can:

By the end of EYFS pupils will be able to use some computational thinking vocabulary and skills, have a basic understanding of online safety/digital life. Use laptops (logging on and accessing age-appropriate websites), iPads, Bee-Bots, Cameras and Remote-control devices.

By the end of KS1 pupils will be able to understand what algorithms are, how they are implemented as programs on digital devices and that programs execute by following precise and unambiguous instructions. Create and debug simple programs. Use logical reasoning to predict the behaviour of simple programs. Use technology purposefully to create, organise, store, manipulate and retrieve digital content. Recognise

common uses of technology beyond school. Use technology safely and respectfully, keeping personal information private; identify where and when appropriate to go for help and support.

By the end of KS2 pupils will be able to design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts. Use sequence, selection, and repetition in programs, work with variables and various forms of input and output. Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs. Understand computer networks including the internet; how they can provide multiple services, such as the world wide web; and the opportunities they offer for communication and collaboration. Use search technologies effectively, appreciate how results are selected and ranked, and be discerning in evaluating digital content. Select, use, and combine a variety of software (including internet services) on a range of digital devices to design and create a range of programs, systems and content that accomplish given goals, including collecting, analysing, evaluating, and presenting data and information. Use technology safely, respectfully, and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact.