Crookham Infant School - An Irresistible Curriculum – Computing

Our School Vision Statement: *At Crookham Infant School we aim to make learning irresistible so that as Team Crookham we all develop the life-long learning habits of resilience, resourcefulness, reflectiveness and reciprocity through exploring together the loving invitation of Jesus to 'live life in all its fullness' John 10:10.*

Our vision stems from our Christian foundation and is firmly rooted in our values: ***Love God, Love Others, Love Learning.***

	Reflective Owl	Resourceful Squirrel	Team Ant	Tough Tortoise
YR	 • *I can tell you what a learner is like* • *I am motivated to explore and find out more* - *I can tell you what I am good at and what I want to get better at* • *I can ask questions and sometimes change my mind in response* • *I can sometimes suggest a solution when talking together to problem solve* • *With support I can plan my learning and make improvements*	• *I know the class routines and am beginning to know what I am learning* • *I can sustain my attention over time especially when I have chosen the task* • *I can select resources for the task or because I am interested in them* • *I can use a range of large and small resources on my own or with others* • *I can talk about what I am learning with some specific vocabulary and listen to other ideas or instructions* 	• *I can learn on my own or with others with just a little support* • *I can work in a pair or group with a little support* • *I am beginning to ignore distractions when I am learning* • *I usually make good choices even when others are distracting* • *I almost always treat others with respect using Kind Words, Kind Hands, Kind Feet* • *I can respond well to other ideas and instructions*	• *I often try myself before asking for help* • *I can focus on some tasks and sustain my involvement* • *I recover from my mistakes and accept some suggestions to change approach,* • *I recover quickly from most disappointments* • *I am beginning to enjoy challenge in my learning (being in the pit)* • *I practise until I master some skills and like to finish I task I start*
Y1	• *I know how I learn best* • *I am motivated to learn for myself and to support others* • *I can ask general questions and decide on an enquiry question* • *I know what I am good at and have an idea of what I can't do yet* • *I use talk to refine my ideas* • *I can plan my learning before I start* • *I can respond to ideas to help me improve my learning*	• *I can use tools (like a visual timetable) to know what I am learning and what is coming next* • *With a little support I can prepare for my learning and make good use of learning time* • *I can select the resource I need for a task and use them safely* • *I can use a wider range of learning resources and vocabulary specific to a subject* • *I listen to others and to instructions*	• *I can usually decide when it is better to work on my own or with others* • *I can usually choose a good partner or group to learn with* • *I understand that sometimes I am the leader and sometimes not* • *I can ignore most distractions and usually make good choices* • *I understand other people's feelings and use Kind Words, Hands, Feet* • *I almost always respond with a good choice*	• *When I start a task I can focus and stay on track* • *I have some strategies to problem solve in my learning* • *I learn from mistakes and can cope with some disappointment* • *I often like to challenge myself and work hard to get out of the learning pit* • *I am getting better at practising until I master new skills and take pride in finishing a task well.*
Y2	• *I always take responsibility for my own learning* • *I am highly motivated to learn and master new things* • *I ask a number of enquiry questions before refining my enquiry focus* • *I can explain in detail what I can and can't do yet* • *I can explain precisely what I am finding difficult in my learning* • *I use talk well to reflect and refine my thinking* • *I make a detailed learning plan* • *I edit and improve before needing support*	• *I always make sure I know what I am learning now and what is next* • *I am well prepared for my learning* • *I am selective over the resources I use choosing what is appropriate for the task* • *I am confident using a range of subject specific resources and vocabulary* • *I am responsible and safe with resources using Kind Hands* • *I know that people are a great resource and listen carefully to ideas and instructions*	• *I make great decisions about when to work independently or collaboratively* • *I can recognise what makes a good learning partner* • *I work well in a pair or a group where I can cope with being the leader or not the leader* • *I can ignore distractions and make independent choices* • *I treat others with respect using Kind Words, Hands and Feet* • *I always listen to others' ideas and instructions and respond appropriately*	• *I can start my learning independently* • *I maintain great focus and sustain it over time* • *I have developed strategies which help my learning* • *I learn from mistakes and can cope with disappointment* • *I challenge myself in the learning pit and expect to work hard* • *I have high expectations of myself and enjoy achieving well* • *I keep practising, make edits and improve work* • *I take pride in the learning journey as well as my finished outcomes*

Curriculum Intent for Computing at Crookham Infant School

The curriculum at Crookham Infant School is designed so that all pupils develop the learning habits of reflection, resourcefulness, reciprocity and resilience. They learn to apply these characteristics of learning effectively across all subjects and curriculum areas whilst at Infant School and continue to develop them in lifelong learning.

The Computing Curriculum at Crookham Infant School uses as its source the highly respected Teach Computing Curriculum designed by the Raspberry Pi Foundation on behalf of the National Centre for Computing Education NCCE. The resources are tailored to the needs of our pupils at Crookham Infant School. The curriculum materials are suitable for all pupils irrespective of their skills, background, and additional needs. The curriculum intentions specific to Computing are:

- *To ensure that Computing is taught with breadth and depth – pupils and teachers understand that Computing is more than programming!*
- *To teach high quality Computing lessons and units based on the latest relevant research*
- *To ensure teachers of Computing have both strong subject knowledge and pedagogy.*

The curriculum is designed with these principles in mind:

Inclusive and ambitious – Irresistible Learning and Developing Life-Long Learning Habits

The Teach Computing Curriculum has been written to support all pupils. Each lesson is sequenced so that it builds on the learning from the previous lesson, and where appropriate, activities are scaffolded so that all pupils can succeed and thrive. Scaffolded activities provide pupils with extra resources, such as visual prompts, to reach the same learning goals as the rest of the class. Exploratory tasks foster a deeper understanding of a concept, encouraging pupils to apply their learning in different contexts and make connections with other learning experiences. As well as scaffolded activities, embedded within the lessons are a range of pedagogical strategies, which support making computing topics more accessible.

Research-informed - Developing Life-Long Learning Habits in order to have 'fullness of life'

The subject of computing is much younger than many other subjects, and as such, there is still a lot more to learn about how to teach it effectively. To ensure that teachers are as prepared as possible, the Teach Computing Curriculum builds on a set of pedagogical principles which are underpinned by the latest computing research, to demonstrate effective pedagogical strategies throughout. To remain up-to-date as research continues to develop, every aspect of the Teach Computing Curriculum is reviewed each year and changes are made as necessary.

Curriculum Design and Implementation

The curriculum is taught in units of learning. Lessons within a unit are taught in a particular order in order to ensure coherence although teachers may teach units within a year group in any order, often linking these units to learning in other areas of the curriculum that term or half-term. The programming units are the exception to this – these are taught in a specific order as concepts and skills rely on prior learning and experiences. The curriculum is carefully planned to ensure coverage of the National Curriculum objectives for EYFS and KS1 and to ensure progression so that pupils know more, recall more and can do more.

All learning outcomes can be described through a high-level taxonomy of ten strands which continue into the KS2 curriculum. They are ordered alphabetically as follows:

- *Algorithms — Be able to comprehend, design, create, and evaluate algorithms*
- *Computer networks — Understand how networks can be used to retrieve and share information, and how they come with associated risks*
- *Computer systems — Understand what a computer is, and how its constituent parts function together as a whole*

- *Creating media — Select and create a range of media including text, images, sounds, and video*
- *Data and information — Understand how data is stored, organised, and used to represent real-world artefacts and scenarios*
- *Design and development — Understand the activities involved in planning, creating, and evaluating computing artefacts*
- *Effective use of tools — Use software tools to support computing work*
- *Impact of technology — Understand how individuals, systems, and society as a whole interact with computer systems*
- *Programming — Create software to allow computers to solve problems*
- *Safety and security — Understand risks when using technology, and how to protect individuals and systems*

The taxonomy provides categories and an organised view of content to encapsulate the discipline of computing. Whilst all strands are present at all phases, they are not always taught explicitly.

The units taught at Crookham Infant School form a spiral curriculum. Each of the themes is revisited regularly (at least once in each year group) with our class Floor Books providing an effective recall and retrieval prompt for all pupils. Our pupils revisit each theme through a new unit that consolidates and builds on prior learning within that theme. This style of curriculum design reduces the amount of knowledge lost through forgetting, as topics are revisited yearly. It also ensures that connections are made even if different teachers are teaching the units within a theme in consecutive years.

***Online Safety** is taught both as part of the Computing curriculum and as part of a wider programme of Personal, Social, Emotional and Health Education. We ensure that the national curriculum requirements for Online Safety are taught by highlighting key objectives within the Computing units and making explicit links to our work in PSHE.*

Progression across Key Stages

All learning objectives are mapped to the NCCE's taxonomy of ten strands, which ensures that units build on each other from one Key Stage to the next. This way we can be sure that our Crookham Infant pupils are well prepared for the move to Junior School in Year 3.

Progression across year groups

Within the Crookham Teach Computing Curriculum every year group learns through units within the same four themes. These themes combine the ten strands of the NCCE taxonomy. The four themes are Computing Systems (and Networks), Programming, Data and Information, Creating Media.

We have carefully planned the Year R learning to prepare children well for the KS1 units of learning and the Year 2 units to prepare children for Junior School. This spiral curriculum approach progresses the skills and concepts effectively from one year group to the next.

Primary Theme	***Computing Systems and Networks***	***Programming***	***Data and Information***	***Creating Media***
Taxonomy Strands	*Computer Systems* *Computer Networks*	*Programming* *Algorithms* *Design and development*	*Data and Information*	*Creating Media* *Design and development*
	Effective use of tools			
	Impact of technology			
	Safety and Security			

Progression within a unit – Learning Graphs

Learning graphs unique to each unit show progression through concepts and skills. Learning intentions form learning milestones and success criteria statements provide smaller steps towards these milestones. This supports teacher assessment of all pupils within each unit of learning.

Pedagogy

High quality teaching and learning in Computing relies on both Subject Knowledge and Pedagogy. At Crookham Infant School we ensure effective lessons for all pupils by using the NCCE approach of 12 key principles of pedagogy as these are underpinned by research and have proven effectiveness. The principles are:

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

These are explored further here:

Lead with concepts *Support pupils in the acquisition of knowledge, through the use of key concepts, terms, and vocabulary, providing opportunities to build a shared and consistent understanding. Glossaries, concept maps, and displays, along with regular recall and revision, can support this approach.*	***Work together*** *Encourage collaboration, specifically using pair programming and peer instruction, and also structured group tasks. Working together stimulates classroom dialogue, articulation of concepts, and development of shared understanding.*	***Get hands-on*** *Use physical computing and making activities that offer tactile and sensory experiences to enhance learning. Combining electronics and programming with arts and crafts (especially through exploratory projects) provides pupils with a creative, engaging context to explore and apply computing concepts.*	***Unplug, unpack, repack*** *Teach new concepts by first unpacking complex terms and ideas, exploring these ideas in unplugged and familiar contexts, then repacking this new understanding into the original concept. This approach, called 'semantic waves', can help pupils develop a secure understanding of complex concepts*	***Model everything*** *Model processes or practices — everything from debugging code to binary number conversions — using techniques such as worked examples and live coding. Modelling is particularly beneficial to novices, providing scaffolding that can be gradually taken away*	***Foster program*** *comprehension Use a variety of activities to consolidate knowledge and understanding of the function and structure of programs, including debugging, tracing, and Parson's Problems. Regular comprehension activities will help secure understanding and build connections with new knowledge.*
Create projects *Use project-based learning activities to provide pupils with the opportunity to apply and consolidate their*	***Add variety*** *Provide activities with different levels of direction, scaffolding, and support that promote learning, ranging from*	***Challenge misconceptions*** *Use formative questioning to uncover misconceptions and adapt teaching to address them as they occur.*	***Make concrete*** *Bring abstract concepts to life with real-world, contextual examples, and a focus on interdependencies*	***Structure lessons*** *Use supportive frameworks when planning lessons, such as PRIMM (Predict, Run, Investigate, Modify, Make)*	***Read and explore code first*** *When teaching programming, focus first on code 'reading' activities, before code writing. With both block-based and*

knowledge and understanding. Design is an important, often overlooked aspect of computing. Pupils can consider how to develop an artefact for a particular user or function, and evaluate it against a set of criteria	*highly structured to more exploratory tasks. Adapting your instruction to suit different objectives will help keep all pupils engaged and encourage greater independence.*	*Awareness of common misconceptions alongside discussion, concept mapping, peer instruction, or simple quizzes can help identify areas of confusion.*	*with other curriculum subjects. This can be achieved through the use of unplugged activities, proposing analogies, storytelling around concepts, and finding examples of the concepts in pupils' lives.*	*and (Use-Modify-Create). These frameworks are based on research and ensure that differentiation can be built in at various stages of the lesson.*	*text-based programming, encourage pupils to review and interpret blocks of code. Research has shown that being able to read, trace, and explain code augments pupils' ability to write code*

Assessment

Within every lesson there are planned opportunities for formative assessment and to address misconceptions promptly. Assessment is largely though observation, questioning, feedback, peer or self-assessment and marking of completed tasks. Through careful assessment teachers adapt their teaching to the needs of all pupils. At the end of each unit of learning teachers will make a summative judgement of which children are on track, working below or working above based on how well the children have met the intended learning outcomes. Floor Books are a key assessment tool for capturing how well children achieve in Computing and engage with Computing within the subject and across the curriculum.

Curriculum Impact of Computing at Crookham Infant School

Our pupils enjoy their learning in Computing and know what to do if they don't feel safe online. Their skills and knowledge are built on over time, within each unit of work and across each year group. They are articulate in reflecting on what they can do more off or know more about using our Floor Books as a prompt for celebrating this achievement.

Structure of the Units of Work in KS1

	Computing Systems (and Networks in KS2)	**Creating Media**	**Programming A**	**Data and Information**	**Creating Media**	**Programming B**
Year R	R.1 Technology in the World around us	R.2 Digital Paintings	R.3 Making things move	Sorting and classifying	Digital symbols	Programs and Animations
Year 1	Technology around us 1.1	Digital Painting 1.2	Moving a Robot 1.3	Grouping Data 1.4	Digital Writing 1.5	Programming and animations 1.6
Year 2	Information Technology around us 2.1	Digital Photography 2.2	Robot Algorithms 2.3	Pictograms 2.4	Digital Music 2.5	Programming Quizzes 2.6

Unit Summaries

Year R	Talk about the lives of the people around them and their roles in society; PP1 Talk about using devices safely and know what to do when not feeling safe.	Begin to explore paint tools. Look at pictures and images.	*Explore using programmable toys and remote controllers. Turn equipment on and off.*	*Sort objects in different ways, e.g. by colour or shape. Begin to find information for example in books and pictures with support*	*Begin to use text such as writing their name or as part of a game*	Begin to engage with objects that can be programmed and with simple animations.
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.	**Moving a robot** Writing short algorithms and programs for floor robots, and predicting program outcomes	**Grouping data** Exploring object labels, then using them to sort and group objects by properties.	**Digital writing** Using a computer to create and format text, before comparing to writing non-digitally.	**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.	**Digital photography** Capturing and changing digital photographs for different purposes.	**Robot algorithms** Creating and debugging programs, and using logical reasoning to make predictions.	**Pictograms** Collecting data in tally charts and using attributes to organise and present data on a computer.	**Digital music** Using a computer as a tool to explore rhythms and melodies, before creating a musical composition.	**Programming quizzes** Designing algorithms and programs that use events to trigger sequences of code to make an interactive quiz.