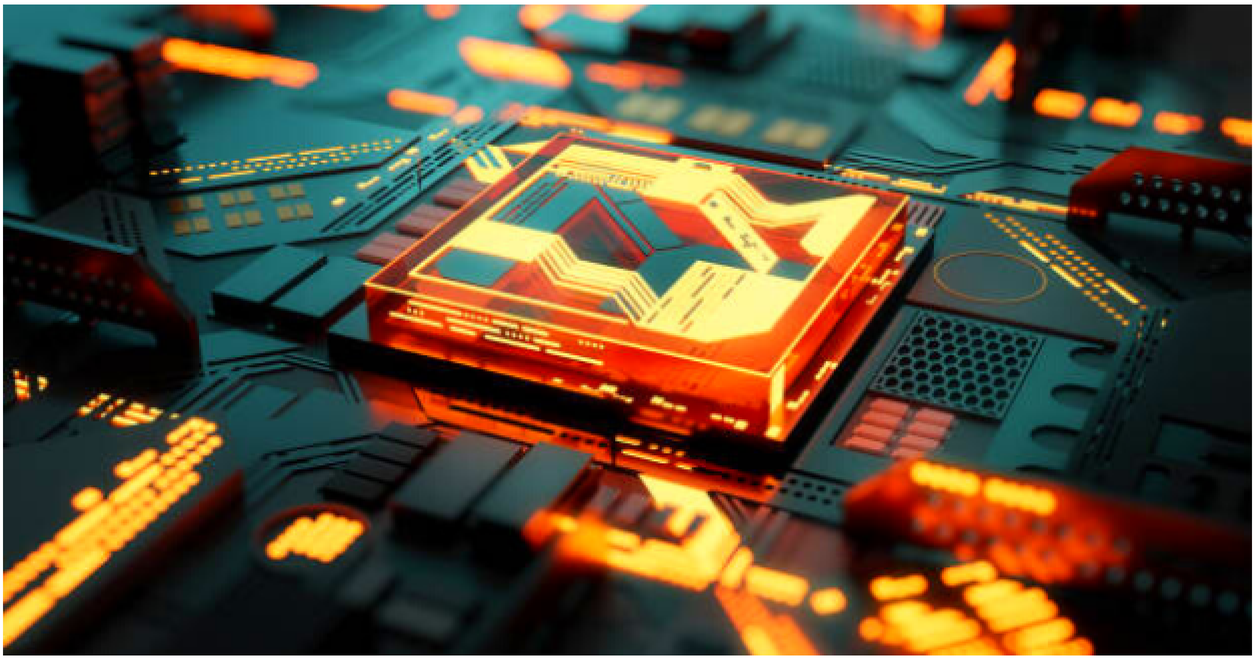
Rosehill Junior School

Computing Policy

January 2023



Last updated: 4th January 2023

Computing Policy Contents

Statement of Intent

At Rosehill, we recognise that understanding Computing is an essential component for functioning within the modern world. Almost everything we do on a daily basis relies on IT in some way shape or form and it is our duty not only to prepare our children to function efficiently today but also prepare them for Tomorrow.

We hope to harness the natural curiosity of our children and encourage them to think creatively in their work, helping them to become problem solvers and innovators. Through a rigorous curriculum offer, we strive to provide the skills, knowledge and understanding of both physical and virtual technologies that our learners require. Whilst allowing them to explore existing and emerging technologies, we aim to ensure they can make informed choices and understand how to keep themselves and their data safe.

Statement of Implementation

Our curriculum is ambitious and rigorous as we follow the National Curriculum 2014 whilst also utilising the current expertise of the National Centre for Computing Education by following the Teach Computing Curriculum. The NCCE is funded by the Department for Education and supporting partners, which marks a significant investment in improving the provision of computing education in England. The NCCE is run by a consortium made up of STEM Learning, the Raspberry Pi Foundation and BCS, The Chartered Institute for IT.

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.

Teaching units for key stages 1 and 2 are based on a spiral curriculum. This means that each of the themes is revisited regularly (at least once in each year group), and 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.

Within the Teach Computing Curriculum, every yeargroup learns through units within the same four themes which combine the ten strands of the NCCE's taxonomy. All learning objectives have been mapped to this taxonomy which ensures that units build on each other from one key stage to the next.

Across each year group, there are 6 sequential units of work to complete, each with 6 lessons covering the 4 themes. 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.

More information on our curriculum and implementation can be found here by reading this document: RJS Computing Curriculum.

An overview of our curriculum coverage and progression can be found by reading this document: RJS Coverage and Progression.

2. Aims and objectives

Purpose of study

A high-quality computing education equips pupils to use computational thinking and creativity to understand and change the world. Computing has deep links with mathematics, science and design and technology, and provides insights into both natural and artificial systems. 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 a level suitable for the future workplace and as active participants in a digital world.

Aims

The national curriculum for computing aims to ensure that all pupils:

- 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

Objectives

Across Key Stage 2, Pupils should be taught 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

3. Curriculum planning

By following the Teach Computing Curriculum, we are meeting the full requirements of the National Curriculum 2014. Learning is sequential and builds upon prior learning. This is further strengthened by the fact that our infant feeder school also follows the same curriculum model so this ensures continuity and progression from KS1.

Coherence and flexibility

The Teach Computing Curriculum is structured in units. For these units to be coherent, the lessons within a unit must be taught in order. However, across a year group, the units themselves do not need to be taught in order, with the exception of 'Programming' units, where concepts and skills rely on prior learning and experiences.

RJS Computing Curriculum Unit Overview

	Computing systems and networks	Creating media	Programming A	Data and information	Creating media	Programming B
Year 3	Connecting computers (3.1)	Stop-frame animation (3.2)	Sequencing sounds (3.3)	Branching databases (3.4)	Desktop publishing (3.5)	Events and actions in programs (3.6)
Year 4	The internet (4.1)	Audio production (4.2)	Repetition in shapes (4.3)	Data logging (4.4)	Photo editing (4.5)	Repetition in games (4.6)
Year 5	Systems and searching (5.1)	Video production (5.2)	Selection in physical computing (5.3)	Flat-file databases (5.4)	Introduction to vector graphics (5.5)	Selection in quizzes (5.6)
Year 6	Communication and collaboration (6.1)	Webpage creation (6.2)	Variables in games (6.3)	Introduction to spreadsheets (6.4)	3D modelling (6.5)	Sensing movement (6.6)

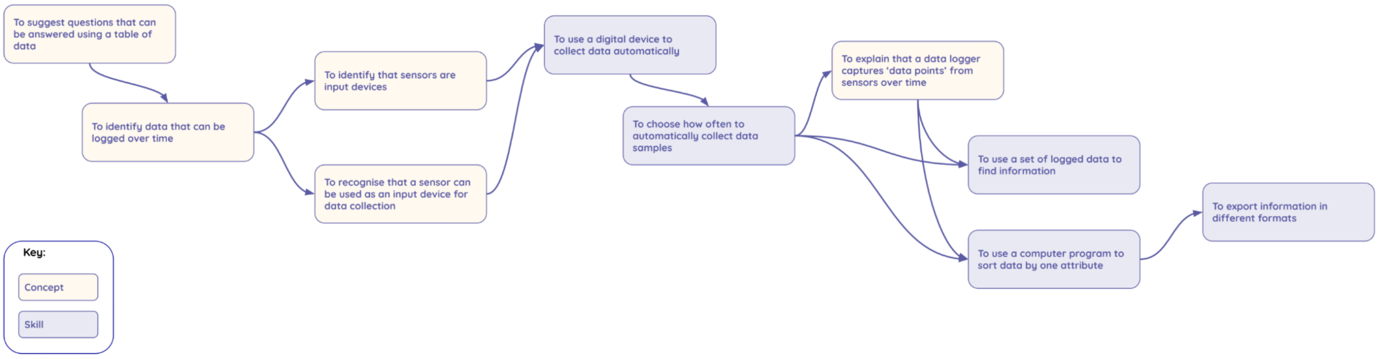
Progression within a unit: learning graphs

Learning graphs are provided as part of each unit and demonstrate progression through concepts and skills. In order to learn some of those concepts and skills, pupils need prior knowledge of others, so the learning graphs show which concepts and skills need to be taught first and which could be taught at a different time. The learning graphs often show more statements than there are learning objectives. All of the skills and concepts learnt are included in the learning graphs. Some of these skills and concepts are milestones, which form learning objectives, while others are smaller steps towards these milestones, which form success criteria. Please note that the wording of the statements may be different in the learning graphs than in the lessons, as the learning graphs are designed for teachers, whereas the learning objectives and success criteria are age-appropriate so that they can be understood by pupils.

In each year group, there are two ‘Programming’ units of work, but only one ‘Programming’ learning graph. The second ‘Programming’ unit builds on the content that was taught in the first ‘Programming’ unit so closely that there is no specific divide where one ends and the other begins.

Learning Graphs are available with each unit of work in the Overview & Assessment Folder from the RJS Curriculum Hub by following the links available under the section headed: RJS Teach Computing Units 2023.

KS2 Example learning graph
Year 4 - Data and Information - Data logging


To suggest questions that can be answered using a table of data
To identify data that can be logged over time
To identify that sensors are input devices
To recognise that a sensor can be used as an input device for data collection
To use a digital device to collect data automatically
To choose how often to automatically collect data samples
To explain that a data logger captures 'data points' from sensors over time
To use a set of logged data to find information
To use a computer program to sort data by one attribute
To export information in different formats
Key:
Concept
Skill

Full details on progression can be seen here: RJS Coverage and Progression.

Knowledge organisation

The Teach Computing Curriculum uses the National Centre for Computing Education's computing taxonomy to ensure comprehensive coverage of the subject. This has been developed through a thorough review of the KS1–4 computing programme of study, and the GCSE and A level computer science specifications across all awarding bodies. All learning outcomes can be described through a high-level taxonomy of ten strands, ordered alphabetically.

National Centre for Computing Education's taxonomy.

- 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

All learning objectives have been mapped to the National Centre for Computing Education's taxonomy of ten strands, which ensures that units build on each other from one key stage to the next. 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 4 reoccurring themes are:

- Computing systems and networks
- Creating media
- Programming
- Data and information

4. Teaching and Learning

Computing is a broad discipline, and computing teachers require a range of strategies to deliver effective lessons to their pupils. The National Centre for Computing Education's pedagogical approach consists of 12 key principles underpinned by research. Each principle has been shown to contribute to effective teaching and learning in computing.

Teachers should use their professional judgement to review, select, and apply relevant strategies for their pupils and examples of their application can be found throughout the units of work at every key stage.

Pedagogical Approach

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 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.

Add variety
Provide activities with different levels of direction, scaffolding, and support that promote learning, ranging from highly structured to more exploratory tasks. Adapting your instruction to suit different objectives will help keep all pupils engaged and encourage greater independence.

Challenge misconceptions
Use formative questioning to uncover misconceptions and adapt teaching to address them as they occur. Awareness of common misconceptions alongside discussion, concept mapping, peer instruction, or simple quizzes can help identify areas of confusion.

Make concrete
Bring abstract concepts to life with real-world, contextual examples and a focus on interdependencies 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.

Structure lessons

Use supportive frameworks when planning lessons, such as PRIMM (Predict, Run, Investigate, Modify, Make) and (Use-Modify-Create). These frameworks are based on research and ensure that differentiation can be built in at various stages of the lesson.

Read and explore code first

When teaching programming, focus first on code 'reading' activities, before code writing. With both block-based and 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.

5. Safeguarding & Online Safety

The unit overviews for each unit show the links between the content of the lessons and the national curriculum and Education for a Connected World framework (ncce.io/efacw). These references have been provided to show where aspects relating to online safety, or digital citizenship, are covered within the Teach Computing Curriculum. Not all of the objectives in the Education for a Connected World framework are covered in the Teach Computing Curriculum, as some are better suited to personal, social, health, and economic (PSHE) education; spiritual, moral, social, and cultural (SMSC) development; and citizenship. However, the coverage required for the computing national curriculum is provided.

Online Safety is also delivered through assemblies on a regular basis and during specified windows such as on Safer Internet Day (this year, 7th of February 2023) or other dates identified on the e-Safety Calendar 22/23. Where appropriate, it is also covered within our PSHE/SMSC lessons and on an ad hoc basis as and when topical incidents occur.

This policy should be read alongside our Safeguarding and e-Safety Policies in addition to the wider resources linked below.

RJS e-Safety Policy

Teaching online safety in school

Education for a Connected World framework

RJS Safeguarding Policy

Childnet Online Safety Calendar

Use of technology in the classroom

A wide range of technology is used during lessons. Prior to using any websites, tools, apps or other online platforms in the classroom, or recommending that pupils use these platforms at home, the class teacher always reviews and evaluates the resource.

Pupils are supervised when using online materials during lesson time – this supervision is suitable to their age and ability.

6. Assessment

Formative assessment

Every lesson includes formative assessment opportunities for teachers to use. These opportunities
are listed in lesson plans and are included to ensure that misconceptions are recognised and addressed if they occur. They vary from teacher observation or questioning, to marked activities.

These assessments are vital to ensure that teachers are adapting their teaching to suit the needs of the pupils that they are working with, and you are encouraged to change parts of the lesson, such as how much time you spend on a specific activity, in response to these assessments.

The learning objective and success criteria are introduced in the NCCE slides at the beginning of every lesson. At the end of every lesson, pupils are invited to assess how well they feel they have met the learning objective using thumbs up, thumbs sideways, or thumbs down. This gives pupils
Assessment a reminder of the content that has been covered, as well as a chance to reflect. It is also a chance for teachers to see how confident the class is feeling so that they can make changes to subsequent lessons accordingly.

Summative assessment

Every unit includes an optional summative assessment framework in the form of either a multiple choice quiz (MCQ) or a rubric. All units are designed to cover both skills and concepts from across the computing national curriculum. Units that focus more on conceptual development include an MCQ. Units that focus more on skills development end with a project and include a rubric. However, within the 'Programming' units, the assessment framework (MCQ or rubric) has been selected on a best-fit basis.

Multiple choice quiz (MCQ)

Each of the MCQ questions has been carefully chosen to represent learning that should have been achieved within the unit. In writing the MCQs, we have followed the diagnostic assessment approach to ensure that the assessment of the unit is useful to determine both how well pupils have understood the content, and what pupils have misunderstood, if they have not achieved

as expected. Each MCQ includes an answer sheet that highlights the misconceptions that pupils may have if they have chosen a wrong answer. This ensures that teachers know which areas to return to in later units.

Rubric

The rubric is a tool to help teachers assess project-based work. Each rubric covers the application of skills that have been directly taught across the unit, and highlights to teachers whether the pupil is approaching (emerging), achieving (expected), or exceeding the expectations for their age group. It allows teachers to assess projects that pupils have created, focussing on the appropriate application of computing skills and concepts. Pedagogically, we want to ensure that we are assessing pupils' understanding of computing concepts and skills, as opposed to their reading and writing skills. This has been carefully considered both in how MCQs have been written (considerations such as the language used, the cultural experiences referenced, etc) and in the skills expected to be demonstrated in the rubric.

Outcomes & Reporting

Teachers will also consider pupil outcomes from tasks set when making judgements on ARE standards. Further Monitoring and Quality Assurance is carried out through lesson observations, learning walks, pupil voice and book scrutiny.

As there are no nationally agreed levels of assessment, any assessment materials are designed to support teaching. The summative assessment will inform teacher judgements around what a pupil has understood in each computing unit, and will feed into the school's assessment process, to align with the whole school approach to assessment in foundation subjects.

7. Subject Leader Roles & Responsibilities

The following responsibilities are carried out by the Computing Subject Leader:

- Ensuring the consistent implementation of Computing policy
- Ensuring continuity between year groups

- Overseeing health and safety policy and practice
- Resources budget management
- Arranging in-service support
- Leading the development and implementation of the school's e-safety policy in line with other Child Protection policies
- Presenting exemplary practice in the teaching of Computing
- Advising colleagues on planning, delivering and assessing Computing
- Monitoring the effective use of technology and giving advice where appropriate
- Ensuring progression in Computing
- Suggested purchasing plans for hardware and software
- Organising Computing resources
- Identifying what support / CPD is needed by individual staff / groups of staff / the whole school
- Reviewing and revising the Computing policy and other associated documents
- Creation of a school portfolio of evidence (if applicable)
- Co-ordinating and overseeing equipment maintenance

Monitoring

The Computing Subject Leader follows a systematic and regular programme of evaluation and monitoring of the Computing curriculum, across the school. This is so that they can monitor the quality of education being provided to all pupils, including:

- Checking that the school's curriculum 'Implementation' matches its 'Intent'
- Evaluating the success (or otherwise) of curriculum planning and delivery
- Having an awareness of impact and be able to demonstrate progression and attainment
- Having an overview of resource and staff training needs

Monitoring is completed via a variety of methods including:

- Observations
- Collecting and analysing planning

- Work scrutinies
- Gathering information from observations of other subjects
- Pupil interviews / pupils voice
- Staff interviews / feedback

As a result of monitoring, appropriate CPD opportunities are provided for staff on an individual, group and whole school basis in line with the school's wider CPD policy, School Development Plan and Strategic Technology Development Plan.

Responsibilities carried out by an ICT Support Technician

All equipment is supported and maintained through a weekly regular visit from a technician who works under the direction of the Computing Subject Leader and Headteacher. The technician reports to the Computing Lead and where appropriate, to the School Business Manager or directly to the Headteacher.

Safe Disposal of Equipment

Government regulations state that any old electrical or electronic equipment must be disposed of in an environmentally responsible way. The regulations which govern this are the Waste Electrical and Electronic Equipment Regulations (WEEE) 2006 and 2013. Schools are therefore required to have a compliant process for disposing of waste electronic and electrical equipment (anything that requires batteries or a plug to operate). School is fully compliant with these regulations and any policy dictated by the Local Authority.

Health and safety

Both staff and children are aware of the need for health and safety to be kept in mind when using technology. All pupils are taught to handle equipment correctly and to switch computers on and off using the correct procedures. The dangers of electricity are stressed and all of the above are presented so as to ensure the pupils respect the equipment and respect other people's work on the computer.

Policy Review

The school recognises that the online world is constantly changing; therefore, the Computing Lead, ICT technicians and the headteacher conduct regular light-touch reviews of this policy to evaluate its effectiveness.

The next scheduled review date for this policy is January 2023.

Signed by:

Adrian Hayes ____________ Headteacher Date: 4th January 2023 ____________

Chris Oxer ____________ Chair of Governors Date: ____________