

Computer Science

I am a coder because I am developing an understanding of instructions, logic and sequences that helps digital technology to run properly

Computing Curriculum **BIG IDEAS**



Information Technology

I am a designer because I use a wide range of APPS and platforms to create and communicate ideas

Digital Literacy

I know that to be a computer scientist or engineer, I need to connect with others safely and respectfully, understanding the need to act within the law



Computing Curriculum—to become a digital engineer

Computing teaching is underpinned by the three BIG IDEAS: **Computer Science, Information Technology and Digital Literacy.**

To become a Digital Engineer, the Curriculum at Nanstallon will help children develop the following characteristics:

- ◆ Competence in coding for a variety of practical and inventive purposes, including the application of ideas within other subjects;
- ◆ The ability to connect with others safely and respectfully, understanding the need to act within the law and with moral and ethical integrity;
- ◆ An understanding of the connected nature of devices;
- ◆ The ability to communicate ideas well by using applications and devices throughout the curriculum;
- ◆ The ability to collect, organise and manipulate data effectively;

To develop these characteristics they will explore the following key concepts:

- | | |
|--------------|---|
| Code: | This concept involves developing an understanding of instructions, logic and sequences. |
| Connect: | This concept involves developing an understanding of how to safely connect with others. |
| Communicate: | This concept involves using apps to communicate one's ideas. |
| Collect: | This concept involves developing an understanding of databases and their uses. |



Computing Curriculum

BIG IDEA	EYFS	KS1	KS2
Computer Science	<ul style="list-style-type: none"> ◇ Be confident to try new activities and show independence, resilience and perseverance in the face of challenge. ◇ Develop their small motor skills so that they can use a range of tools competently, safely and confidently. 	<p>Understanding Algorithms:</p> <ul style="list-style-type: none"> ◇ Recognize and use basic algorithms and sequencing in simple programs. ◇ Develop logical thinking skills through unplugged activities and simple programming tools. <p>Programming Concepts:</p> <ul style="list-style-type: none"> ◇ Use visual programming languages to create simple programs. ◇ Understand and use basic programming concepts like loops and conditions. 	<p>Computational Thinking:</p> <ul style="list-style-type: none"> ◇ Apply computational thinking skills to solve problems and break them down into smaller steps. ◇ Understand algorithms, variables, and debugging techniques. <p>Programming and Coding:</p> <ul style="list-style-type: none"> ◇ Create and debug programs using block-based and text-based programming languages. ◇ Explore concepts such as loops, conditionals, variables, and functions.
Information Technology	<ul style="list-style-type: none"> ◇ Safely use and explore a variety of materials, tools and techniques, experimenting with colour, design, texture, form and function. ◇ Explore, use and refine a variety of artistic effects to express their ideas and feelings. 	<p>Computer Basics:</p> <ul style="list-style-type: none"> ◇ Identify and name common hardware components of a computer. ◇ Understand the purpose of input and output devices. <p>Using Software and Applications:</p> <ul style="list-style-type: none"> ◇ Navigate and use age-appropriate software and applications for specific purposes. ◇ Develop basic skills in word processing, image editing, and multimedia tools. 	<p>Data and Information:</p> <ul style="list-style-type: none"> ◇ Understand how data is represented and stored digitally. ◇ Interpret and present data using charts, graphs, and spreadsheets. <p>Multimedia Creation:</p> <ul style="list-style-type: none"> ◇ Use digital tools and software to create and edit multimedia content like images, videos, and presentations. ◇ Understand basic principles of design, layout, and accessibility. <p>Networks and Communication:</p> <ul style="list-style-type: none"> ◇ Understand the basics of computer networks, including the internet. ◇ Communicate, collaborate, and share information using digital platforms.
Digital Literacy	<ul style="list-style-type: none"> ◇ Know and talk about the different factors that support their overall health and wellbeing: sensible amounts of 'screen time'. ◇ Explain the reasons for rules, know right from wrong and try to behave accordingly. 	<p>Online Safety and Responsibility:</p> <ul style="list-style-type: none"> ◇ Recognize and understand the importance of online safety and responsible use of technology. ◇ Identify and respond to potential online risks and dangers. <p>Basic Digital Skills:</p> <ul style="list-style-type: none"> ◇ Use digital tools and devices to communicate, create, and present simple content. ◇ Understand file management, saving, and organizing digital files. 	<p>Online Safety and Cybersecurity:</p> <ul style="list-style-type: none"> ◇ Understand and apply strategies to stay safe online, including responsible social media use and privacy settings. ◇ Recognize the importance of data protection and cybersecurity. <p>Digital Research and Evaluation:</p> <ul style="list-style-type: none"> ◇ Use digital tools to gather information, critically evaluate sources, and present findings. ◇ Understand copyright, plagiarism, and intellectual property rights.



Computing Curriculum

Milestones

MILESTONE 1

Key Stage 1 (Ages 5-7):

Computer Science:

1. Understanding Algorithms:

Recognize and use basic algorithms and sequencing in simple programs.
Develop logical thinking skills through unplugged activities and simple programming tools.

2. Programming Concepts:

Use visual programming languages to create simple programs.
Understand and use basic programming concepts like loops and conditions.

Digital Literacy:

1. Online Safety and Responsibility:

Recognize and understand the importance of online safety and responsible use of technology.
Identify and respond to potential online risks and dangers.

2. Basic Digital Skills:

Use digital tools and devices to communicate, create, and present simple content.
Understand file management, saving, and organizing digital files.

Information Technology:

1. Computer Basics:

Identify and name common hardware components of a computer.
Understand the purpose of input and output devices.

2. Using Software and Applications:

- Navigate and use age-appropriate software and applications for specific purposes.

Develop basic skills in word processing, image editing, and multimedia tools.

MILESTONE 2

Key Stage 2 (Ages 7-11):

Computer Science:

1. Computational Thinking:

Apply computational thinking skills to solve problems and break them down into smaller steps.
Understand algorithms, variables, and debugging techniques.

2. Programming and Coding:

Create and debug programs using block-based and text-based programming languages.
Explore concepts such as loops, conditionals, variables, and functions.

Digital Literacy:

1. Online Safety and Cybersecurity:

Understand and apply strategies to stay safe online, including responsible social media use and privacy settings.
Recognize the importance of data protection and cybersecurity.

2. Digital Research and Evaluation:

Use digital tools to gather information, critically evaluate sources, and present findings.
Understand copyright, plagiarism, and intellectual property rights.

Information Technology:

1. Data and Information:

Understand how data is represented and stored digitally.
Interpret and present data using charts, graphs, and spreadsheets.

2. Multimedia Creation:

Use digital tools and software to create and edit multimedia content like images, videos, and presentations.
Understand basic principles of design, layout, and accessibility.

3. Networks and Communication:

Understand the basics of computer networks, including the internet.
Communicate, collaborate, and share information using digital platforms.



Computing Curriculum

Milestones

Assessment and Progression:

Assessment in computing will focus on pupils' understanding, application of skills, and ability to solve problems. This can be done through practical tasks, projects, and formative assessments. Progression will be evident as pupils demonstrate increased confidence, computational thinking skills, and the ability to create more complex programs and projects.

Conclusion:

The primary school computing curriculum provides a foundation for pupils to develop their digital literacy, computational thinking, and information technology skills. It aims to nurture responsible and critical users of technology, preparing them for the digital world they will navigate in the future. By providing a comprehensive and engaging computing curriculum, we empower our pupils to become confident and capable participants in the digital age.



Computing Curriculum

Progression

Remember to adapt these activities based on your specific curriculum requirements and the needs and abilities of your pupils

<u>Year 1:</u>	<p>Introduction to coding: Use age-appropriate coding platforms like ScratchJr or Bee-Bot to introduce basic programming concepts through block-based coding.</p> <p>Digital storytelling: Create simple digital stories using software like Book Creator or PowerPoint, incorporating images,</p>
<u>Year 2:</u>	<p>Online safety: Teach pupils about internet safety, including topics like personal information protection, safe browsing, and responsible online behaviour.</p> <p>Introduction to algorithms: Help pupils understand algorithms by creating step-by-step instructions for everyday tasks,</p>
<u>Year 3:</u>	<p>Basic HTML coding: Introduce pupils to the basics of HTML coding by teaching them how to create a simple webpage using tags, headings, images, and links.</p> <p>Data representation: Teach pupils about different types of data and ways to represent them, such as bar graphs or pic-</p>
<u>Year 4:</u>	<p>Introduction to spreadsheets: Introduce pupils to spreadsheet software like Microsoft Excel or Google Sheets. Teach them how to enter data, perform basic calculations, and create simple charts.</p> <p>Internet research skills: Teach pupils how to search effectively on the internet, evaluate the credibility of online</p>
<u>Year 5:</u>	<p>Digital citizenship: Explore topics like online etiquette, cyberbullying, and responsible social media use. Encourage discussions and collaborative activities around digital citizenship.</p> <p>Introduction to programming: Introduce text-based programming languages like Python or JavaScript using kid-friendly coding platforms. Teach basic programming concepts like variables, loops, and conditionals.</p>
<u>Year 6:</u>	<p>Website design: Teach pupils how to plan and design a website using tools like Wix or Weebly. Discuss elements of good web design, such as layout, navigation, and accessibility.</p> <p>Data analysis and visualization: Guide pupils in analysing real-world data using spreadsheet software. Help them create advanced charts and graphs to present their findings.</p>



Computing Curriculum

Coding and programming activity ideas for each year group (years 1-6) in a primary school computing curriculum:

Year 1:

- Sequence with Code Cards: Create a set of code cards with simple instructions (e.g., move forward, turn left, turn right). Pupils use the code cards to create sequences and guide a classmate or a programmable robot through a maze or obstacle course.
- Bee-Bot Shapes: Use a programmable robot like Bee-Bot to introduce basic geometry. Pupils program the robot to move and create different shapes on a grid.
- Introduce children to code.org to help familiarise them with the the basic ideas of coding.
- Gradually work through Course A in a semi-structured way to ensure all children are able to understand and move at their own pace.
- Thinking about how the Great Fire of London started, create an algorithm for something that happened in the bakery. This might be making a sandwich, baking bread or maybe a cake. Type up these algorithms on the computer and make the connection between instructions and computer code.
- Use the iPads to learn about Photography and how we can explore the outside world using technology.
- Set the children a series of nature-themed scavenger hunts where they need to go and photograph plants and flowers that meet certain descriptors.
- Label a photograph of a plant to show the different parts. Using iPads, we will enhance the area around us by photographing and then drawing over images of the school site to add more nature through plants, flowers and animals.
- Revisit coding on code.org before moving onto using Scratch Jnr on iPads to explore hands-on coding experiences.
- Discuss how we can write instructions that can have an affect on other things and how a computer won't do anything on its own – it needs to be told what to do.



Computing Curriculum

Coding and programming activity ideas for each year group (years 1-6) in a primary school computing curriculum:

Year 2:

- **Animated Stories:** Use block-based coding platforms like ScratchJr or Scratch to create animated stories. Pupils can code characters, backgrounds, and movements to bring their stories to life.
- **Algorithms with Dance:** Have pupils create step-by-step algorithms for simple dance moves. They can code a sequence of dance moves and perform them, following their own algorithms.
- **Coding with code.org.** Start children on Course B and work through with whole class input and individual student-led learning. Highlight and discuss key concepts as we arrive at them and ensure pupils have a good understanding of what key concepts and ideas mean.
- **Build a 2D Castle in Keynote** using blocks, textures and shapes. Learn about how we can adjust, resize and move objects on a computer and how to copy/paste.
- **Photograph a place** from around the school and then use this as a background for the castles.
- **Use Shapes in Keynote** to create different styles of tessellation. Learn about Copy + Paste, Grouping, Rotating etc. Create a honeycomb image using tessellation and bees.
- **Explore pollinators and pollination** using Minecraft Education Edition to become a bee and learn how to create honey.
- **Use the iPads** to create a rich document full of Bee Facts and illustrations, using hexagons to link facts together.
- **Use the internet safely** to research information and use the iPads and Crayons to draw images.
- **Discuss recycling** and what can and can't be recycled. Create a visual document to represent waste being sent to landfill by filling a pyramid shape with non-recyclable objects.
- **Staying Safe Online.** Discuss how we use the internet and what risks there might be.
- **With the iPads** we can start to create informational posters to remind us how to avoid these dangers and stay safe online



Computing Curriculum

Coding and programming activity ideas for each year group (years 1-6) in a primary school computing curriculum:

Year 3:

- Interactive Quiz: Pupils create an interactive quiz using Scratch or other block-based coding tools. They code questions and feedback, allowing their classmates to test their knowledge on a specific topic.
- Coding Challenges: Introduce coding challenges that involve problem-solving and logical thinking. For example, ask pupils to code a program that draws different geometric shapes or calculates simple math problems.
- Use Tinkercad to design and create 3D Christmas Decorations which will then be printed on 3D Printer before the end of term. Encourage children to think of bold, interesting designs that work well with the split contrast colour schemes.
- Online Safety with Be Internet Legends. Introduce the five areas of BIL and begin to have honest and open discussions about how we use the internet and how to ensure we stay safe online.
- Discuss what the internet is and how it works - and why this can be both a good and a bad thing in terms of how we stay safe.
- Hour of Code: Minecraft Edition Reintroduce Scratch and begin to explore different blocks of code and what we can build with them.
- Learn about algorithms, repeating loops and more.
- Connect Micro:Bits to Scratch and learn about how we can use these as input and output devices. Create a series of programs which use Input, Output and then Sensors to affect the code on screen. Discuss how sensors can be used to measure changes in the world around us, such as in the rainforest.
- Use Green screen app to create an Augmented Reality exploration through a rainforest.



Computing Curriculum

Coding and programming activity ideas for each year group (years 1-6) in a primary school computing curriculum:

Year 4:

- Maze Game: Pupils design and code a maze game using Scratch or a similar platform. They create a character that needs to navigate through a maze, avoiding obstacles and collecting points.
- Animated Digital Greeting Card: Using a coding tool like Scratch, pupils code an animated digital greeting card for a specific occasion, such as a birthday or holiday.
- Create an animation of a Roman Temple in Keynote by learning about shapes, symbols and Magic Move.
- Research facts about a Roman Emperor and add these to the animations we created.
- Online Safety with Be Internet Legends. Follow on from the learning started in Year 3 by taking a deeper dive into the five areas of Be Internet Legends and encourage rich, honest conversations about how these might affect us when we're using the internet.
- Use Scratch to invent, create and program robots to fulfil a series of challenges. These robot inventions will complete certain activities using Scratch coding.
- Use the iPads to explore Drawing and Photography.
- Create a piece of Art Text and a Photo Collage around the theme of inventions and the future.
- Use iMovie or Clips to record a News Report from the scene of either an Earthquake or an Explosion. Research facts online and write a script.
- Use Green Screen special effects.
- Use Keynote to create an interactive app which shares information about volcanoes and earthquakes around the world. Focus on facts and data linked to each location. Discuss how to research data, images etc and how you know what information is reliable and what isn't.



Computing Curriculum

Coding and programming activity ideas for each year group (years 1-6) in a primary school computing curriculum:

Year 5:

- Interactive Storytelling: Pupils create interactive stories using Scratch or another coding platform. They incorporate choices and branching narratives, allowing users to make decisions that affect the outcome of the story.
- Game Design: Introduce game design principles and have pupils create their own games using a platform like Scratch. They can design characters, levels, and game mechanics.
- How does the Internet work? How does this differ from other methods of communication (i.e. Stone Age times)
- Online Safety with Be Internet Legends.
- Create a Podcast Interview with an Alien. Use the internet to research facts and write a script, then record in GarageBand with Sound Effects and Editing.
- Design Cover Artwork and learn about Importing/Exporting Files.
- Use Scratch to create interactive apps and games to help people make good choices to protect our environment for the future.
- Use Micro:Bits, and Scratch to create programs which use sensors to help us measure noise, temperature etc.



Computing Curriculum

Coding and programming activity ideas for each year group (years 1-6) in a primary school computing curriculum:

Year 6:

- Text-Based Adventure Game: Pupils use a text-based programming language like Python to create an interactive adventure game. They code the storyline, choices, and outcomes using conditional statements and variables.
- App Prototyping: Introduce app design concepts and have pupils create paper or digital prototypes of their own mobile app ideas. They can design user interfaces and map out the functionality of their apps.
- The History of Computing including Codebreaking, Alan Turing and Bletchley Park. Consider how this influences modern computers and encryption.
- What is a computer? Build one from its component parts.
- Use Spike Prime Robotics to create robots that use light in different ways. Use Light Sensors to activate movements. Use iPads to plan and create long exposure images using light trails.
- Online Safety with Be Internet Legends. Recap previous years learning and design and create a '6th Pillar' to teach others about being safe online.
- Design a character, poster and game for this 6th Pillar.
- Remix popular songs and use Live Loops in GarageBand to create our own songs and soundtracks. Begin learning the basics of Video Editing using iMovie and bring in our GarageBand songs as Soundtracks.
- Video Editing and Special Effects. [ECC Video: Special Effects chapter] Learn to use Green Screen, Jump Cuts, Reverse Footage, Stop Motion and other special effects.
- Learn what types of content can and can't be shared safely and learn techniques to share in a more safe way.



Computing Curriculum

Assessment of Disciplinary Knowledge

These questions tease out the disciplinary knowledge within each subject – and should work when discussing any block of learning completed in the unit. How does each lesson contribute to being able answer the disciplinary questions?

KEY STAGE 1

1. What application or software am I using?
2. What hardware am I using?
3. Why am I using this software / application / hardware?
4. How is it similar to other application / software / hardware I've used before? How is it different?
5. What do I want to achieve by using this technology?

KEY STAGE 2

6. What is the purpose of what I am doing?
7. Who is going to use what I'm doing?
8. What is the best application or software to use to achieve my intended purpose?
9. What is the best hardware to use to achieve my intended purpose?
10. What is the most efficient and effective way to achieve my intended purpose?



Computing Curriculum

END POINTS

Progression in computing happens over a period of time. Aligned to the 7 Qi (non-cognitive) skills, the statements below are intended to represent what our aspiration of a Y6 digital engineer will be, having had the breadth and depth of learning experiences on their primary journey.

ME	WILL	WHAT IF?	WHY?
<p>"Pupils will explore and identify their personal interests and strengths in the field of computing, fostering self-awareness and a sense of purpose."</p> <p>Pupils will be encouraged to reflect on their favourite aspects of technology, such as game design or website development, helping them understand their personal interests within the broader field of computing.</p>	<p>"Pupils will learn to set goals and create plans for their computing projects, developing problem-solving and project management skills."</p> <p>Before starting a coding project, pupils will create a project plan outlining the objectives, timeline, and milestones they aim to achieve, teaching them how to set and work towards goals effectively.</p>	<p>"Pupils will engage in 'What If' scenarios, encouraging them to explore the possibilities and potential of technology."</p> <p>Pupils will be presented with hypothetical scenarios like "What if you could design a new app for your school?" This will prompt them to think creatively and explore the limitless possibilities that technology offers.</p>	<p>"Pupils will analyse the 'why' behind computing concepts, fostering critical thinking and inquisitiveness about technology."</p> <p>When learning about algorithms, pupils will be prompted to question why certain algorithms are more efficient than others, encouraging critical thinking and a deeper understanding of computational principles.</p>

WE	WIGGLE	WOBBLE
<p>"Pupils will collaborate with their peers on various computing projects, enhancing teamwork and communication skills."</p> <p>In group coding projects, pupils will work together to design and code a simple game, where they will assign roles (e.g., programmer, designer, tester) and communicate effectively to bring the</p>	<p>"Pupils will be encouraged to 'wiggle' by exploring creative solutions and innovative approaches to computing tasks."</p> <p>When learning about algorithms, pupils will be prompted to question why certain algorithms are more efficient than others, encouraging critical thinking and a deeper understanding of compu-</p>	<p>"Pupils will encounter challenges in coding and problem-solving, encouraging adaptability and resilience in the face of setbacks."</p> <p>During debugging sessions, pupils will face errors in their code and will need to persistently troubleshoot and adapt their approach until the program runs smoothly, fostering resilience in</p>