



# HEYHOUSES C.E. PRIMARY SCHOOL COMPUTING CURRICULUM





**At Heyhouses we aspire to be all that God has created us to be.**

*'I can do all things through Christ who strengthens me.'* Philippians 4:13

Our aim and purpose in education is based on firm beliefs and values; that Jesus is our redeemer; that each individual is unique and valued; and that although all different, we are dependent upon one another.

In our school we seek to provide for the spiritual, mental, moral and physical development, growth and well-being of all our children.

**— Firm Foundations — Ambitious Learning — Flourishing for life —**

# Curriculum Intent



At Heyhouses, we are now following the Teach Computing scheme from years 1 – 6, which will ensure a thorough coverage of the National Curriculum objectives as well as proper progression between our year groups.

Our curriculum is closely mapped to the National Centre for Computing Excellence (NCCE) computing taxonomy, which comprises ten key strands of knowledge and skills for Key Stages 1 and 2.

The ten NCCE computing strands, and their associated learning outcomes, are:

1. **Algorithms** — Be able to comprehend, design, create, and evaluate algorithms;
2. **Computer networks** — Understand how networks can be used to retrieve and share information, and how they come with associated risks;
3. **Computer systems**— Understand what a computer is, and how its constituent parts function together as a whole;
4. **Creating media** — Select and create a range of media including text, images, sounds, and video;
5. **Data and information** — Understand how data is stored, organised, and used to represent real-world artefacts and scenarios;
6. **Design and development** — Understand the activities involved in planning, creating, and evaluating computing artefacts;
7. **Effective use of tools** — Use software tools to support computing work;
8. **Impact of technology** — Understand how individuals, systems, and society as a whole interact with computer systems;
9. **Programming** — Create software to allow computers to solve problems;
10. **Safety and security** — Understand risks when using technology, and how to protect individuals and systems.

[Teach Computing Curriculum Journey](#)



# Curriculum Intent



## Spiral Curriculum

The 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.

## Physical Computing

The Teach Computing Curriculum acknowledges that physical computing plays an important role in modern pedagogical approaches in computing, both as a tool to engage pupils and as a strategy to develop pupils' understanding in more creative ways. Additionally, physical computing supports and engages a diverse range of pupils in tangible and challenging tasks.

The physical computing units in the Teach Computing Curriculum are:

- Year 5 – Selection in physical computing, which uses a Crumble controller
- Year 6 – Sensing movement, which uses a micro:bit

## Online Safety

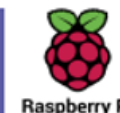
The unit overviews for each unit show the links between the content of the lessons and the national curriculum and Education for a Connected World framework ([ncce.io/efacw](https://www.ncce.io/efacw)). These references have been provided to show where aspects relating to online safety, or digital citizenship, are covered within the Teach Computing Curriculum. Further learning in this area is provided in personal, social, health, and economic (PSHE) and the wider curriculum. The coverage required for the computing national curriculum is provided.



# Computing Curriculum



	<b>Autumn 1 Computer Systems and Networks</b>	<b>Autumn 2 Creating Media</b>	<b>Spring 1 Programming A</b>	<b>Spring 2 Data and Information</b>	<b>Summer 1 Creating Media</b>	<b>Summer 2 Programming B</b>
<b>Year 1</b>	<a href="#">Technology around us</a> Recognising technology in school and using it responsibly.	<a href="#">Digital painting</a> Choosing appropriate tools in a program to create art, and making comparisons with working non-digitally.	<a href="#">Moving a robot</a> Writing short algorithms and programs for floor robots, and predicting program outcomes.	<a href="#">Grouping data</a> Exploring object labels, then using them to sort and group objects by properties.	<a href="#">Digital writing</a> Using a computer to create and format text, before comparing to writing non-digitally.	<a href="#">Programming animations</a> Designing and programming the movement of a character on screen to tell stories.
<b>Year 2</b>	<a href="#">Information technology around us</a> Identifying IT and how its responsible use improves our world in school and beyond.	<a href="#">Digital photography</a> Capturing and changing digital photographs for different purposes.	<a href="#">Robot algorithms</a> Creating and debugging programs, and using logical reasoning to make predictions.	<a href="#">Pictograms</a> Collecting data in tally charts and using attributes to organise and present data on a computer.	<a href="#">Digital music</a> Using a computer as a tool to explore rhythms and melodies, before creating a musical composition.	<a href="#">Programming quizzes</a> Designing algorithms and programs that use events to trigger sequences of code to make an interactive quiz.
<b>Year 3</b>	<a href="#">Connecting computers</a> Identifying that digital devices have inputs, processes,	<a href="#">Stop-frame animation</a> Capturing and editing digital still images to	<a href="#">Sequencing sounds</a> Creating sequences in a block-based	<a href="#">Branching databases</a> Building and using	<a href="#">Desktop publishing</a> Creating documents by modifying text,	<a href="#">Events and actions in programs</a> Writing algorithms and programs that



# Computing Curriculum



	and outputs, and how devices can be connected to make networks.	produce a stop-frame animation that tells a story.	programming language to make music.	branching databases to group objects using yes/no questions.	images and page layouts for a specified purpose.	use a range of events to trigger sequences of actions.
<b>Year 4</b>	<a href="#">The internet</a> Recognising the internet as a network of networks including the WWW, and why we should evaluate online content.	<a href="#">Audio Production</a> Capturing and editing audio to produce a podcast, ensuring that copyright is considered.	<a href="#">Repetition in Shapes</a> Using a text based programming language to explore count-controlled loops when drawing shapes.	<a href="#">Data Logging</a> Recognising how and why data is collected over time, before using data loggers to carry out an investigation.	<a href="#">Photo Editing</a> Manipulating digital images, and reflecting on the impact of changes and whether the required purpose is fulfilled.	<a href="#">Repetition in Games</a> Using a block-based programming language to explore count controlled and infinite loops when creating a game.
<b>Year 5</b>	<a href="#">Systems and searching</a> Recognising IT systems in the world and how some can enable searching on the internet.	<a href="#">Video production</a> Planning, capturing, and editing video to produce a short film.	<a href="#">Selection in physical computing</a> Exploring conditions and selection using a programmable microcontroller.	<a href="#">Flat-file databases</a> Using a database to order data and create charts to answer questions.	<a href="#">Introduction to vector graphics</a> Creating images in a drawing program by using layers and groups of objects.	<a href="#">Selection in quizzes</a> Exploring selection in programming to design and code an interactive quiz



# Computing Curriculum

<b>Year 6</b>	<u><a href="#">Communication and collaboration</a></u> Exploring how data is transferred by working collaboratively online.	<u><a href="#">Webpage creation</a></u> Designing and creating webpages, giving consideration to copyright, aesthetics, and navigation.	<u><a href="#">Variables in games</a></u> Exploring variables when designing and coding a game.	<u><a href="#">Introduction to spreadsheets</a></u> Answering questions by using spreadsheets to organise and calculate data.	<u><a href="#">3D modelling</a></u> Planning, developing, and evaluating 3D computer models of physical objects.	<u><a href="#">Sensing movement</a></u> Designing and coding a project that captures inputs from a physical device.
---------------	--------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------

This resource is available online at [nccpe.io/tcc](https://nccpe.io/tcc).

This resource is licensed under the Open Government Licence, version 3. For more information on this licence, see [nccpe.io/ogl](https://nccpe.io/ogl).

Contributions: We would like to thank the many people who helped to create the Teach Computing Curriculum: our content writers, advisors, reviewers, pilot schools, and every teacher who has taken the time to send us feedback.

