Teach Computing

Teacher G

ey Stage 2

guide to the Teach Computing Curriculum

Teach Computing

ontents

troduction

Teach Computing Curriculum (ncce.io/tcc) is a
prehensive collection of materials produced to support
hours of teaching, facilitating the delivery of the entire
ish computing curriculum from key stage 1 to 4
16-year-olds). The Teach Computing Curriculum was
ted by the Raspberry Pi Foundation on behalf of the
onal Centre for Computing Education (NCCE). All
ent is free, and editable under the Open Government
nce (OGL — ncce.io/ogl), ensuring that the resources
e tailored to each individual teacher and school
ng. The materials are suitable for all pupils irrespective
eir skills, background, and additional needs.

The aims of the Teach Computing Curriculu
as follows:

- Reduce teacher workload
- Show the breadth and depth of the comp
  curriculum, particularly beyond program
- Demonstrate how computing can be tau
  on research
- Highlight areas for subject knowledge ar
  enhancement through training

The Teach Computing Curriculum resource
updated in response to feedback. Feedbac
submitted at ncce.io/rrfeedback or by ema
resourcesfeedback@raspberrypi.org.

Teach Computing

urriculum design

e approach

ıerence and flexibility

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

ıwledge organisation

Teach Computing Curriculum uses the National Centre omputing Education's computing taxonomy to ensure orehensive coverage of the subject. This has been loped through a thorough review of the KS1–4 outing programme of study, and the GCSE and A level outer science specifications across all awarding es. All learning outcomes can be described through h-level taxonomy of ten strands, ordered abetically as follows:

- **Algorithms** — Be able to comprehend, d and evaluate algorithms
- **Computer networks** — Understand how be used to retrieve and share informatio they come with associated risks
- **Computer systems** — Understand what a how its constituent parts function togeth
- **Creating media** — Select and create a ra including text, images, sounds, and vide
- **Data and information** — Understand hov organised, and used to represent real-w and scenarios
- **Design and development** — Understand involved in planning, creating, and evalu computing artefacts
- **Effective use of tools** — Use software t computing work
- **Impact of technology** — Understand hov systems, and society as a whole interac computer systems

Teach Computing

ral curriculum

units for key stages 1 and 2 are based on a spiral
culum. This means that each of the themes is
ited regularly (at least once in each year group),
pupils revisit each theme through a new unit that
olidates and builds on prior learning within that
e.

style of curriculum design reduces the amount of
/ledge lost through forgetting, as topics are revisited
ly. It also ensures that connections are made even
erent teachers are teaching the units within a theme
nsecutive years.

Physical computing

The Teach Computing Curriculum acknowl
physical computing plays an important role
pedagogical approaches in computing, bot
to engage pupils and as a strategy to deve
understanding in more creative ways. Addi
computing supports and engages a diverse
in tangible and challenging tasks.

The physical computing units in the Teach C
Curriculum are:

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

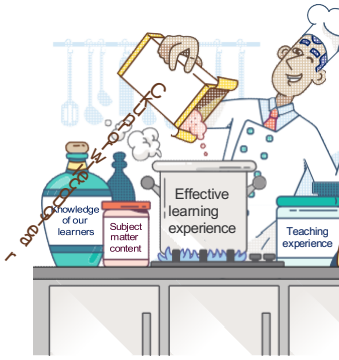
Teach Computing

re principles

usive and ambitious

Teach Computing Curriculum has been written to ort all pupils. Each lesson is sequenced so that it s on the learning from the previous lesson, and where opriate, activities are scaffolded so that all pupils can eed and thrive. Scaffolded activities provide pupils extra resources, such as visual prompts, to reach the e learning goals as the rest of the class. Exploratory s foster a deeper understanding of a concept, uraging pupils to apply their learning in different exts and make connections with other learning riences.

ell as scaffolded activities, embedded within the ns are a range of pedagogical strategies (defined in Pedagogy' section of this document), which support ng computing topics more accessible.



tructure of the units of work

y unit of work in the Teach Computing Curriculum contains: a unit overview; a learning graph, to show th
ding a detailed lesson plan, slides for learners, and all the resources you will need; and formative and s

ach Computing Curriculum overview

	Computing systems and networks	Creating media	Programming A
Year 3	Connecting computers (3.1)	Stop-frame animation (3.2)	Sequencing sounds (3.3)
Year 4	The internet (4.1)	Audio editing (4.2)	Repetition in shapes (4.3)
Year 5	Sharing information (5.1)	Video editing (5.2)	Selection in physical computing (5.3)
Year 6	Internet communication (6.1)	Webpage creation (6.2)	Variables in games (6.3)

Teach Computing

Structu

t summaries

	Computing systems and networks	Creating media	Programming A
Year 3	**Connecting computers** Identifying that digital devices have inputs, processes, and outputs, and how devices can be connected to make networks.	**Stop-frame animation** Capturing and editing digital still images to produce a stop-frame animation that tells a story.	**Sequencing sounds** Creating sequences in a block-based programming language to make music.
Year 4	**The internet** Recognising the internet as a network of networks including the WWW, and why we should evaluate online content.	**Audio editing** Capturing and editing audio to produce a podcast, ensuring that copyright is considered.	**Repetition in shapes** Using a text-based programming language to explore count-controlled loops when drawing shapes.

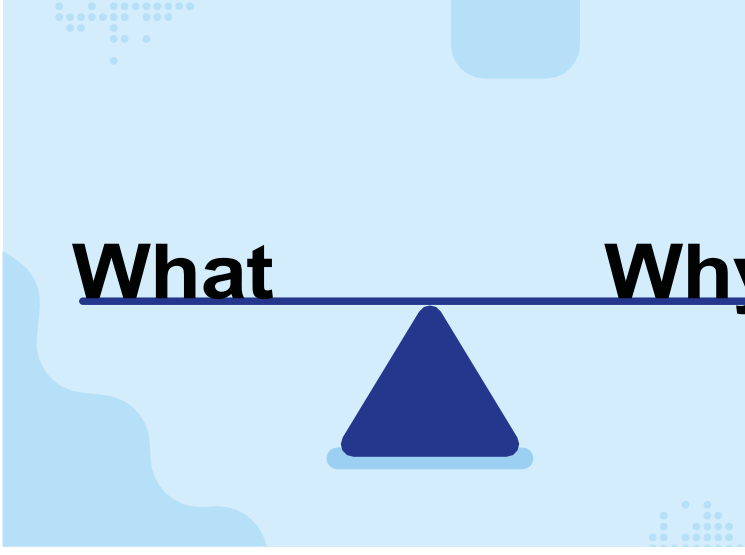
Teach Computing

t summaries

	Computing systems and networks	Creating media	Programming A
Year 5	**Sharing information** Identifying and exploring how information is shared between digital systems.	**Video editing** Planning, capturing, and editing video to produce a short film.	**Selection in physical computing** Exploring conditions and selection using a programmable microcontroller.
Year 6	**Internet communication** Recognising how the WWW can be used to communicate and be searched to find information.	**Webpage creation** Designing and creating webpages, giving consideration to copyright, aesthetics, and navigation.	**Variables in games** Exploring variables when designing and coding a game.

National Curriculum Coverage — Years 3 and 4	3.1 Connecting
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	

National Curriculum Coverage — Years 5 and 6	5.1 Sharing information
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	✓


What
Why

Teach Computing

rogression

ogression across key stages

earning objectives have been mapped to the National re for Computing Education's taxonomy of ten strands, h ensures that units build on each other from one key e to the next.

ogression across year groups

in the Teach Computing Curriculum, every year group ns through units within the same four themes, which bine the ten strands of the National Centre for puting Education's taxonomy (see table, right).

approach allows us to use the spiral curriculum oach (see the 'Spiral curriculum' section for more mation) to progress skills and concepts from one group to the next.

Primary themes	Computing syst and networ
Taxonomy strands	Computer systems Computer networks

Teach Computing

›gression within a unit — ırning graphs

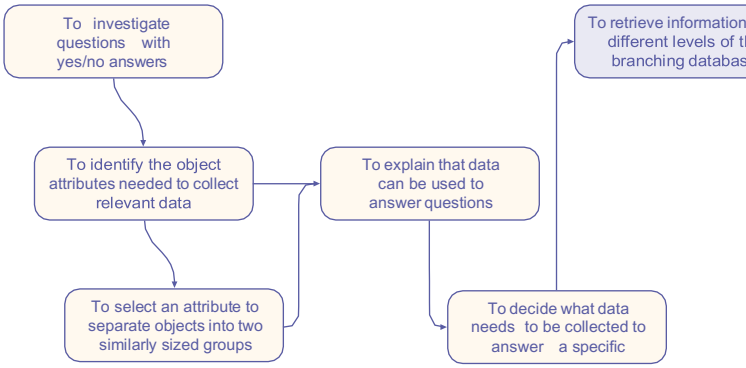
ning graphs are provided as part of each unit demonstrate progression through concepts and s. In order to learn some of those concepts and s, pupils need prior knowledge of others, so the ning graphs show which concepts and skills d to be taught first and which could be taught different time.

learning graphs often show more statements there are learning objectives. All of the skills and epts learnt are included in the learning graphs. e of these skills and concepts are milestones, h form learning objectives, while others are ler steps towards these milestones, which form ess criteria. Please note that the wording of the ments may be different in the learning graphs in the lessons, as the learning graphs are gned for teachers, whereas the learning ctives and success criteria are age-appropriate at they can be understood by pupils.

In each year group, there are two ‘Progra units of work, but only one ‘Programming’ graph. The second ‘Programming’ unit bui

Teach Computing

- Branching databases – learning graph



Teach Computing

edagogy

puting is a broad discipline, and computing teachers ire a range of strategies to deliver effective lessons eir pupils. The National Centre for Computing cation's pedagogical approach consists of 12 key ciples underpinned by research: each principle has shown to contribute to effective teaching and ning in computing.

ecommended that computing teachers use their essional judgement to review, select, and apply ant strategies for their pupils.

e 12 principles are embodied by the Teach puting Curriculum, and examples of their application e found throughout the units of work at every key e. Beyond delivering these units, you can learn more t these principles and related strategies in the nal Centre for Computing Education pedagogy it (ncce.io/pedagogy).

Lead with concepts

Support pupils in the acquisition of knowled use of key concepts, terms, and vocabulary opportunities to build a shared and consisten Glossaries, concept maps (ncce.io/qr07), an with regular recall and revision, can suppor

Structure lessons

Use supportive frameworks when planning such as PRIMM (Predict, Run, Investigate, — ncce.io/qr11) and Use-Modify-Create. Th are based on research and ensure that di be built in at various stages of the lesson

Make concrete

Bring abstract concepts to life with real-wo examples and a focus on interdependenci curriculum subjects. This can be achieved of unplugged activities, proposing analogi around concepts, and finding examples of in pupils' lives.

Teach Computing

d and explore code first

n teaching programming, focus first on code 'reading' ities, before code writing. With both block-based and based programming, encourage pupils to review and pret blocks of code. Research has shown that being to read, trace, and explain code augments pupils' y to write code.

ate projects

project-based learning activities to provide pupils with pportunity to apply and consolidate their knowledge understanding. Design is an important, often looked aspect of computing. Pupils can consider to develop an artefact for a particular user or function, evaluate it against a set of criteria.

del everything

el processes or practices — everything from gging code to binary number conversions — using niques such as worked examples (ncce.io/qr02) ive coding (ncce.io/qr05). Modelling is particularly ficial to novices, providing scaffolding that can adually taken away.

Get hands-on

Use physical computing and making activit tactile and sensory experiences to enhanc Combining electronics and programming w crafts (especially through exploratory projec pupils with a creative, engaging context to apply computing concepts.

Challenge misconceptions

Use formative questioning to uncover misc and adapt teaching to address them as th Awareness of common misconceptions al discussion, concept mapping, peer instruc simple quizzes can help identify areas of co

Add variety

Provide activities with different levels of di scaffolding, and support that promote acti ranging from highly structured to more expl Adapting your instruction to suit different c help keep all pupils engaged and encourag independence.

Teach Computing

ssessment

rmative assessment

y lesson includes formative assessment
rtunities for teachers to use. These opportunities
sted in the lesson plan and are included to ensure
misconceptions are recognised and addressed
y occur. They vary from teacher observation
estioning, to marked activities.

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

learning objective and success criteria are introduced
e slides at the beginning of every lesson. At the end
ery lesson, pupils are invited to assess how well they
hey have met the learning objective using thumbs
humbs sideways, or thumbs down. This gives pupils
a reminder of the content that has been co
as a chance to reflect. It is also a chance f
see how confident the class is feeling so t
make changes to subsequent lessons acc

Summative assessment

Every unit includes an optional summative
framework in the form of either a multiple
(MCQ) or a rubric. All units are designed t
skills and concepts from across the compu
curriculum. Units that focus more on conc
development include an MCQ. Units that f
on skills development end with a project a
a rubric. However, within the 'Programming
assessment framework (MCQ or rubric) ha
selected on a best-fit basis.

Teach Computing

age group. It allows teachers to assess projects
upils have created, focussing on the appropriate
cation of computing skills and concepts.

gogically, we want to ensure that we are assessing
s' understanding of computing concepts and skills,
posed to their reading and writing skills. This has
carefully considered both in how MCQs have been
en (considerations such as the language used, the
ral experiences referenced, etc) and in the skills
cted to be demonstrated in the rubric.

apting for your setting

ere are no nationally agreed levels of assessment,
ssessment materials provided are designed to be
and adapted by schools in a way that best suits
needs. The summative assessment materials will
m teacher judgements around what a pupil has
rstood in each computing unit, and could feed into
hool's assessment process, to align with their
oach to assessment in other foundation subjects.

Teach Computing

esources

ftware and hardware

puting is intrinsically linked to technology and efore requires that pupils experience and use ıge of digital tools and devices. As the Teach puting Curriculum was being written, careful ideration was given to the hardware and software cted for the units. The primary consideration was we felt a tool would best allow pupils to meet ıing objectives; the learning always came first the tool second.

ıake the units of work more accessible to pupils and ners, the materials include screenshots, videos, and uctions, and these are based on the tools listed in able below. The list below should not be seen as xplicit requirement for schools. Schools may choose e alternative tools that offer the same features as ribed in the units. All of the learning objectives can ıet with alternative hardware and software, as the ıing objectives are not designed to be tool-specific.

Software

If you do not wish to use the software reco the units, you could use an alternative piec that provides the same function. All learnir should be achievable using alternative sof however, there will be a lot less support fo as screenshots and demonstration videos software referenced in the materials.

The units of work include the use of free s would need to be installed on local machi software that is available as an online too software needs to be installed locally, sc need to plan software installation in adva

Several of the units that use online tools re to sign up to free services in order to acce This also allows pupils the opportunity to s projects that they are working on, and give skills that they need to manage their own u passwords as digital citizens. However, the

Teach Computing

tware and hardware overview

ıirements for pupils — below

	Desktop or laptop	Chromebook	
3.1 Connecting computers	✓	•	
3.2 Stop-frame animation	•	•	
3.3 Sequencing sounds	✓	•	
3.4 Branching databases	✓	•	
3.5 Desktop publishing	✓	•	
3.6 Events and actions in programs	✓	•	
4.1 The internet	✓	•	
4.2 Audio editing	✓		
4.3 Repetition in shapes	✓	•	
4.4 Data logging	✓	+	
4.5 Photo editing	✓	•	
4.6 Repetition in games	✓	•	

Ised for the unit — reflected in screenshots ● Could be used as an alternative ✚ Data loggers that work with Chrom

Teach Computing

tware and hardware overview, cont.

irements for pupils — below

	Desktop or laptop	Chromebook	
5.1 Sharing information	✓	•	
5.2 Video editing	✓	•	
5.3 Selection in physical computing	✓	•	
5.4 Flat-file databases	✓	•	
5.5 Vector drawing	✓	•	
5.6 Selection in quizzes	✓	•	
6.1 Internet communication	✓	•	
6.2 Webpage creation	✓	•	
6.3 Variables in games	✓	•	
6.4 Introduction to spreadsheets	✓	•	
6.5 3D modelling	✓	•	
6.6 Sensing	✓	•	

lsed for the unit — reflected in screenshots • Could be used as an alternative

Teach Computing

ational Centre for Computing Educatio

National Centre for Computing Education (NCCE) nded by the Department for Education and marks nificant investment in improving the provision mputing education in England.

NCCE is run by a consortium made up of STEM ning, the Raspberry Pi Foundation, and BCS, Chartered Institute for IT. Our vision is to achieve rld-leading computing education for every child gland.

The NCCE provides high-quality support fo of computing in schools and colleges, from through to A level. Our extensive range of resources, and support covers elements of at every key stage, catering for all levels o knowledge and experience.

For further information, visit: teachcomputin

resource is available online at ncce.io/tcc.

ributions: We would like to thank the many people who helped to create the Teach Computing Curriculu rs, advisors, reviewers, pilot schools, and every teacher who has taken the time to send us feedback.