Teach Computing

Teacher G

ey Stage 1

guide to the Teach Computing Curriculum

Teach Computing

ontents

Introduction

Curriculum design

The approach
- Coherence and flexibility
- Knowledge organisation
- Spiral curriculum
- Physical computing
- Online safety

Core principles
- Inclusive and ambitious
- Research-informed
- Time-saving for teachers

Structure of the units of work

5 Teach Computing Curriculum overvie
5 Brief overview
6 Unit summaries
7 National curriculum coverage – ke computing curriculum
8 Teaching order
8 Mixed year groups

9 Progression

9 Progression across key stages
9 Progression across year groups
10 Progression within a unit — learning

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 (5- to
ear-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
be 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. Feedba
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 ɔrehensive coverage of the subject. This has been loped through a thorough review of the KS1–4 ɔuting programme of study, and the GCSE and A level ɔuter 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 informati 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 vid
- **Data and information** — Understand ho organised, and used to represent real-v and scenarios
- **Design and development** — Understanc involved in planning, creating, and eval computing artefacts
- **Effective use of tools** — Use software t computing work
- **Impact of technology** — Understand ho 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 revisited
larly (at least once in each year group), and pupils
it each theme through a new unit that consolidates
builds on prior learning within that theme.

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 if
rent teachers are teaching the units
n a theme in consecutive 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. Addit
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

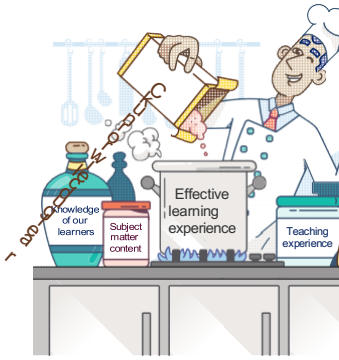
Teach Computing

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.



Teach Computing

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
ent — including a detailed lesson plan, slides for learners, and all the resources you will need; and form

ach Computing Curriculum overview

f overview

	Computing systems and networks[1]	Creating media	Programming A
Year 1	Technology around us (1.1)*	Digital painting (1.2)	Moving a robot (1.3)
Year 2	Information technology around us (2.1)	Digital photography (2.2)	Robot algorithms (2.3)

vorks are not part of the key stage 1 national curriculum for computing but the title is used as a strand a
numbers in the brackets are a 'quick code' reference for each unit, eg 1.3 refers to the third Year 1 unit in

Teach Computing

Struc

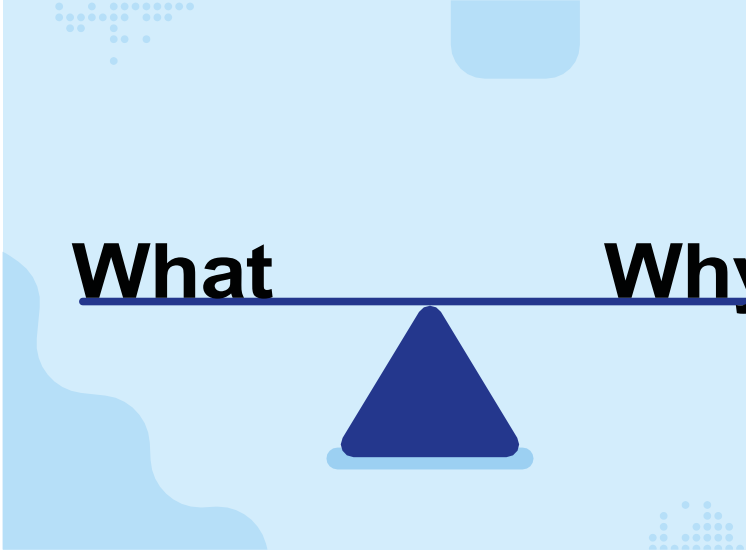
t summaries

	Computing systems and networks	Creating media	Programming A
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.
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.

Teach Computing

Struc

National Curriculum Coverage — Key Stage 1 Computing Curriculum	1.1 Technology around us	
Understand what algorithms are, how they are implemented as programs on digital devices, and that programs execute by following precise and unambiguous instructions		
Create and debug simple programs		
Use logical reasoning to predict the behaviour of simple programs		
Use technology purposefully to create, organise, store, manipulate and retrieve digital content	✓	
Recognise common uses of information technology beyond school	✓	
Use technology safely and respectfully, keeping personal information private; identify where to go for help and support when they have concerns about content or contact on the internet or other online technologies	✓	


What
Why

Teach Computing

rogression

ogression across key stages

arning 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

n the Teach Computing Curriculum, every year group is through units within the same four themes, which oine 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.

themes	Computing syste and networ
Taxonomy strands	Computer systems Computer networks

Teach Computing

›gression within a unit — ırning graphs

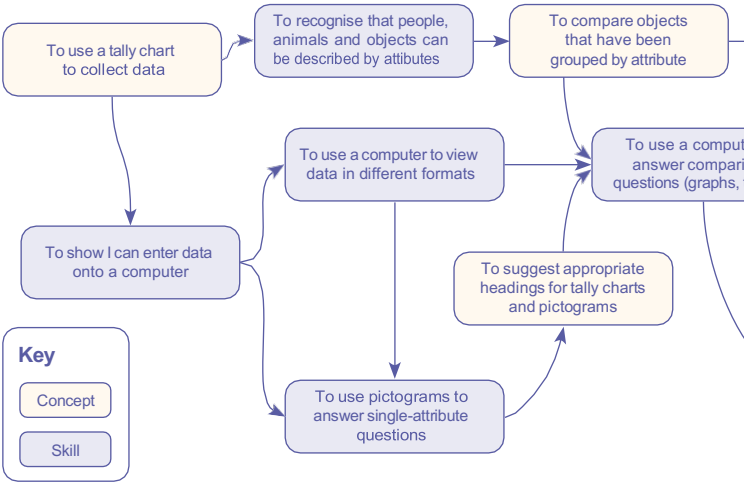
ning graphs are provided as part of each unit demonstrate progression through concepts and ;. In order to learn some of those concepts and ;, pupils need prior knowledge of others, so the ıing graphs show which concepts and skills need › taught first and which could be taught at ferent time.

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

- Data and Information – Pictograms 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 ssional 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 as PRIMM (Predict, Run, Investigate, Modify io/qr11) and Use-Modify-Create. These fra based on research and ensure that differe 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 oret 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 ooked aspect of computing. Pupils can consider how velop an artefact for a particular user or function, and uate 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) and coding (ncce.io/qr05). Modelling is particularly ficial to novices, providing scaffolding that can be ually 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 misco adapt teaching to address them as they oc of common misconceptions alongside disc concept mapping, peer instruction, or simp can help identify areas of confusion.

Add variety

Provide activities with different levels of di scaffolding, and support that promote acti ranging from highly structured to more exp 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 are d in the lesson plan and are included to ensure that onceptions are recognised and addressed if they r. They vary from teacher observation or questioning, arked activities.

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

earning 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

Pedagogically, when we assess, we want we are assessing a pupil's understanding concepts and skills, as opposed to their re writing skills. Therefore, we encourage ob assessment while pupils are still developin skills. We believe that this is the most relia capture an accurate picture of learning.

Observing learning

To capture summative assessment data o we recommend using the success criteria and capturing some of the following while taking place:

Teach Computing

l of the unit

pil working at age-related expectations should be able
et the success criteria for each lesson by the end of the
However, it should also be noted that some pupils may
longer to grasp certain skills and concepts and therefore
achieve a success criterion from a lesson at a later date.

e end of a unit, you may wish to use the observations
/ou have made across each of the lessons to determine
/erall snapshot of a pupil's understanding of the content
that unit.

apting for your setting

ıere are no nationally agreed levels of assessment, the
ssment materials provided are designed to be used
adapted by schools in a way that best suits their needs.
summative assessment materials will inform teacher
ements around what a pupil has understood in each
puting unit, and could feed into a school's assessment
ess, to align with their approach to assessment in other
dation subjects.

Teach Computing

esources

ftware and hardware

puting is intrinsically linked to technology and efore requires that pupils experience and use a range gital tools and devices. As the Teach Computing iculum was being written, careful consideration was n to the hardware and software selected for the units. primary consideration was how we felt a tool would allow pupils to meet learning objectives; the learning ys came first and 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 the below. The list below should not be seen as an cit requirement for schools. Schools may choose to 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 pie that provides the same function. All learni should be achievable using alternative soft there will be a lot less support for teachers screenshots and demonstration videos ref software referenced in the materials.

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

Several of the units that use online tools r 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, th

Teach Computing

tware and hardware overview

ıirements for pupils — below

✓ Us

	Desktop or laptop	Chromebook	
1.1 Technology around us	✓	•	
1.2 Digital painting	✓	•	
1.3 Moving a robot			
1.4 Grouping data	✓	•	
1.5 Digital writing	✓	•	
1.6 Programming animations	•	•	
2.1 Information technology around us	✓	•	
2.2 Digital photography	✓		
2.3 Robot algorithms			
2.4 Pictograms	✓	•	
2.5 Making music	✓	•	
2.6 Programming quizzes	•	•	

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.