Computing Textbook Criteria

Index

About the criteria

These criteria are informed by evidence on the importance of textbooks and the characteristics of high quality textbooks. In high-performing jurisdictions, teachers and pupils are able to use substantial, well-structured paper-based materials which sequence learning and include carefully-structured explanatory text which introduce challenging and essential ideas and language. They support understanding and progression through carefully varied presentation of ideas and through activities which reveal key concepts and develop secure understanding. They include frequent assessment activities which support pupil understanding and teacher judgements about progress. In computing, the very nature of the discipline means that this includes carefully-designed practice and application.

Modern textbooks in high performing systems often involve a ‘suite’ of materials - a pupil textbook, a teacher version of the textbook or a ‘teacher handbook’, practice books, and allied digital resources. The criteria are not unduly restrictive regarding the balance of these, but they emphasise key elements of form and content - leading to substantial, coherent, and engaging materials which present the ‘shape and content’ of computing as a discipline, give support to teachers, and give all pupils access to understanding of the fundamentals of the discipline.

Note on digital materials

This initiative focusses on the characteristics of high quality paper resources. This is because there is strong evidence that a good supply of high quality paper resources has a positive and direct impact on educational quality and reduced teacher workload.

The role which supporting digital materials can assume is recognised, so these paper resources may be accompanied by high quality, supportive, digital resources for the extra opportunities they can offer. Digital materials can differ greatly in form and function, including high quality demonstrations, videos, interactive tasks, environments for development and testing, as well as interaction with educators and other learners and rich learning environments.

Digital materials must be consistent with the models and approaches derived for the paper based materials and the two must be coherent. We do not wish to see digital materials which are simply a cursory ‘add-on’ or addition to paper; they should be an integral part of the overall offer. A specific balance or ‘split’ between paper based materials and digital materials is not advocated and it is for publishers to develop well-evidenced and coherent combinations.

Some high quality digital materials are already available for the Computing curriculum; these have been created by the National Centre for Computing Education.

Criteria a. Product format and digital requirements

1. 'Core' and 'Supplemental' Components	Explanatory Notes
a. A **core component** must be submitted, which must be one of the following: i. a teacher-facing textbook for primary teachers teaching Key Stage 1 of the computing programme of study in the English national curriculum. ii. a pupil-facing textbook for Key Stage 2 pupils, or secondary-aged pupils studying Key Stage 3 and/or Key Stage 4 content of the computing programme of study in the English national curriculum. b. The core component of the submitted product must be paper based (See **Note 1**). c. **Supplemental component(s)** can be included as part of a single product submission providing they complement the core component (See **Note 2**). d. Starter Code practice exercises for the programming part of the curriculum must not be paper-based and must be submitted as a supplemental digital component. (See **Note 3)** e. Submitted products should promote teachers' further use of digital resources beyond what the product itself provides (See **Note 4**). f. Products which cover computing qualifications at KS4 and KS5 are out of scope for these criteria and cannot be accepted.	**Note 1** The panel reserves the right to reject pupil-facing components which are deemed cumbersome/heavy, or which are not user friendly. **Note 2** Supplemental components may be, for example, pupil practice books or online pupil exercises; and may be either paper-based or digital in format. Where digital versions of the paper-based core components are also available, these will not be separately assessed, and will subject to the same outcome as the paper-based version, providing the content is substantially the same. **Note 3** Starter Code practice exercises are a requirement (See **Criteria d.1.b.)** and must be provided in a digital format as a supplemental component. This format might be, for example, a single zip file, or as part of a larger supplemental digital component. **Note 4** Videos, animations, or films, for example, can offer illustrative and stimulating opportunities to enhance pupils' understanding of how computers and the digital world function. Teachers' selection and adaptation of such materials (i.e. those available beyond what is offered by the supplier and the submitted product's core/supplement components) to suit the pupils in their setting should be encouraged.

Criteria b. – Curriculum Coverage

1. Computing curriculum coverage	Explanatory Notes
a. The submitted products must cover the complete subject content of a full Key Stage of the computing curriculum, as a minimum. b. Coverage of the curriculum content topics must be proportionate to the four aims of the computing curriculum. c. Products for primary teachers must offer the option of a **year-by-year route** through the materials (See **Note 1**). d. Key Stage 3/4 teacher-facing components must set out the pre-requisite knowledge needed by pupils before the textbook can be used effectively to support their progress (See **Note 2**). e. Products must: i. make links between computing curriculum topics to ensure that pupils can develop an understanding of how they fit together (See **Note 3**) ii. be clear on the fundamental computing concepts that pupils need to learn, in depth, and cover them in more than one way to solidify learning (See **Note 4)** iii. use contemporary computing and 'real world' examples, scenarios and references across all types of content (See **Note 5**).	**Note 1** The option of a year-by-year presentation of material and progression through National Curriculum content must be offered to support non-specialist teachers, but this should not be the only route on offer. **Note 2** Textbooks may include activities or short assessments to evaluate whether pupils have the necessary pre-requisites for a block of learning on which they are about to embark. **Note 3** Cross-curricular links may also be made to engage the pupil, but must not be enforced or made essential to learning, or detract from focussed learning directed to the content of the National Curriculum. **Note 4** The product may expand or unpack the curriculum content into constituent parts; but, to reduce cognitive overload, fewer items covered in greater depth is preferable to a breadth of items covered in less detail. **Note 5** Examples should: ● reflect current computing practices ● use up to date computing terminology ● make links to current issues, such as environmental sustainability (i.e. energy use by devices) or automation (i.e. self-driving cars) ● not refer to obsolete items or outdated processes, such as fax machines, file transfer protocol, defunct social media platforms.

Criteria c. Teacher Support

1. Teacher Support	Explanatory Notes
a. Teacher-facing components must: i. include an explanation of how the product and any supplementary materials, can enable children to progress from a defined starting point to full understanding of the subject content. ii. explain the product's pedagogical approaches and learning models, referencing the underpinning research that supports it (See **Note 1**), and why different approaches may be used for different concepts. iii. demonstrate a consistent approach to teaching the principles of programming, informed by the findings of relevant research (See **Note 2**). iv. provide advice on recognising common and predictable computing misconceptions and misunderstandings, and how to help pupils overcome these (See **Note 3**)	**Note 1** For example: • use of the Block Model (Schulte, 2008) as a lens through which you can understand program comprehension • evidence of the importance of discussion and reflection, in relation to topics such as the opportunities offered by computer networks and the internet. **Note 2** The approach is expected to balance reading and understanding code with writing new code. **Note 3** Any remedial actions suggested should support pupil progression in their understanding. For programming misconceptions, advice must not be specific to any individual programming language. Non-programming misconceptions might include, for example, pupils' confusion of databases with spreadsheets.

Criteria d. Practice Exercises

1. Practice Exercises	Explanatory Notes
a. Practice exercises must: i. describe the intended learning outcome and how it links to the computing curriculum subject content (See **Note 1**) ii. support pupils in consolidating core knowledge and skills iii. outline what pre-requisite knowledge is needed for the pupil to complete them (if applicable) iv. model a variety of styles and formats for breadth and depth, which are relevant to the subject matter being taught (See **Note 2)** v. including a mixture of unplugged, online and offline activities (See **Note 3**) vi. have due consideration for pupils who speak English as an additional language, and/or those with additional needs (See **Note 4**). b. Practice exercises covering programming must: i. provide pupils with extensive practical experience of designing, creating and debugging computer programs in order to solve problems ii. use, and build on, starter code (See **Note 5**) iii. include questions about the structure and function of a piece of code (See **Note 5**) iv. include worked examples and steps on how to solve a problem, where appropriate v. not focus solely on short programming exercises that start from scratch within the product vi. encourage 'pair programming' where applicable vii. go beyond copying provided code e.g. use of 'Copy Code'.	**Note 1** The publisher should consider where best to situate the exercises, and such accompanying information, within their component(s). **Note 2** Examples of styles/formats which could be included: ● focused 'quick-check' activities ● tightly framed questions that focus pupils' attention on a particular learning point, such as misconceptions ● activities which are more expansive than "quick-check" and require pupil reflection ● written exercises ● repetitive exercises ● exercises which have scope for variability ● multiple choice or identifying missing words. **Note 3** Offline activities might include discussion exercises, role play, or storytelling, to support pupils to master a skill prior to using technology. Unplugged activities help to develop computational thinking and understand concepts without technology, e.g., using jigsaws to design algorithms. **Note 4** Activities may require following detailed written instructions and due regard should be given to ensuring that this is accessible to pupils with additional needs, such as dyslexia or pupils who speak English as an additional language. **Note 5** Starter code can be used to illustrate the use of syntax within set programming tasks and to improve pupil confidence. This should not be done in a way that leads to a requirement for pupils to copy out code from the textbook. Pupils may answer questions about the structure and function of a piece of code, e.g. the 'Investigate' aspect of PRIMM.

Criteria e – Inclusion and supporting all pupils

1. Inclusion	**Explanatory Notes**
a. Product content must cover and represent a wide and varied range of computing applications which can affect everyday life, e.g. people, products and life experiences. (See **Note 1**). b. Product content must not contain 'loaded' terms, e.g., those that could be seen to encourage the use of stereotyping (See **Note 2).** c. Clear and simple language should be used so that it may be easily understood by those who speak English as an additional language. d. Products must not contain teaching approaches which encourage the disclosure of pupils' personal circumstances (See **Note 3**). e. The publisher must take steps to ensure that content in all components is accessible to all users, such as those with visual impairments.	**Note 1** Names, images, and other content should reflect a diverse audience. **Note 2** Examples of loaded terminology frequently appearing within computing would include e.g. "black box testing" and "master/slave". **Note 3** Some activities may unintentionally cause the pupil to reveal disadvantage or may cause distress, for example "All about me", "how many devices in your home", "my family" etc.
2. <u>Supporting all pupils</u>	
a. The product's approach should convey the importance of setting high expectations for every pupil, in an objective and impartial way. b. The product must demonstrate that all pupils, regardless of their current and historical rate of progress, are entitled to appropriate support, stretch and challenge. c. The product must support the progression of all pupils. It must not lead to 'labelling' of pupils as 'less able' or 'low attaining', which can lead to a reduced curriculum for some pupils. Rather, it should support social learning and rich discussion amongst all pupils. It should allow pupils who have acquired more advanced computing skills to be stretched and developed, for example through extension activities and more demanding application (See **Note 4**).	**Note 4** **Examples:** ● the product should recommend a level of activity that allows a novice or 'beginner' to acquire the required levels of knowledge ● recommend an approach that allows for a reasonable time extension where appropriate ● provide extension activities, for those pupils who are ready to progress to the next stage, e.g. which: ○ put knowledge and skills into practice ○ include structured and open-ended exercises ○ challenge pupils to think about a concept further to deepen their understanding.

Criteria f. Assessment

1. Assessment	Explanatory Notes
a. The product must clearly outline the learning objectives associated with each chapter or question, to aid the teacher and/or pupil with assessment. b. The product must encourage an approach that supports formative and summative assessment at the end of each learning chapter or section, to consolidate learning and capture the progress made. c. The product must support an assessment approach that stimulates understanding, through analysing pupil understanding and progress made. d. The assessment approach must enable individuals to be aware of the knowledge that they have acquired and how this links to the wider computing curriculum, through peer review, rich questions and other means, such as hinge questions (See **Note 1**). e. The product must contain relevant and timely questions which are embedded throughout, including the use of rich questions and 'distractors' or incorrect questions (See **Note 2**). f. Teacher-facing products must provide accurate answers, or prompts to look for, for reviewing pupils' responses to assessments. g. The product must support teachers and pupils to identify and rectify common subject misconceptions relating to computing, through the use of a variety of question types.	**Note 1** Definition of 'rich' and 'hinge' questions: ● Rich questions: questions which are focussed on key idea and vital concepts, stimulate pupils' thinking, allows individual and group discourse, stretches thinking but is accessible, and provide insights into thinking and attainment for both pupils and teachers. They can involve a requirement to reflect or work on a series of smaller questions or activities, before they return to the original question'. ● A hinge-point question is a diagnostic question addressed to a whole class which enables the teacher to know what proportion of their class have secured the necessary level of understanding of a particular concept, when deciding whether to recap or move on. **Note 2** Definition of 'distractor questions' ● 'distractor' questions are the incorrect options in a set of multiple-choice questions. High quality distractors are chosen to challenge common areas of misunderstanding.

Criteria g. Coherence and terminology

1. **Content coherence**	**Explanatory Note**
a. The product's content must be laid out with clarity, to aid cognitive load. b. The product must be consistent in style and use common formats throughout when presenting content e.g. for information elements, instructions and questions (See **Note 1**). c. The product must demonstrate coherence across the different types of content, e.g. text, activities, assessment questions, illustrations. d. Each section or chapter of the product must have a clear purpose, prefaced with an introduction. e. When combined, the sections or chapters of the product must have a clear and linear sequence, which, when taken together, form a clear progression of the learning objectives (See **Note 2**). f. The sections or chapters of the product must contain information elements that cohere (See **Note 3**).	**Note 1** This includes use of colour, emboldening/italicising, highlighting keywords, the use of titles/ subtitles and avoiding erroneous references. Text-based programming exercises should use monospaced type. **Note 2** The panel may use the product's contents page to help inform assessment about whether the product has a linear sequence and its constituent parts cohere. **Note 3** Any text, assessment questions, illustrations or other elements qualify as an 'information element' on a page. They should all relate to each other and must have a clear function, e.g. focus on the learning objectives.
2. Terminology	
a. Technical language/terms should be highlighted and must be used consistently throughout the product. b. Technical terms should not be avoided - it is important that technical vocabulary is introduced as appropriate, and its correct use is encouraged. c. Correct and precise computing terminology should be introduced with the concepts, outlining how and when they should be used and applied. d. A full glossary must be provided within the product, outlining the diversity of computing terms where applicable, e.g. when specific terms are used for different purposes (See **Note 4**) e. For pupil-facing material, aids should be provided to assist with cognitive load, such as "cheat" or "crib" sheets and/or reference guides.	**Note 4** Examples: • some computing terms have a different definition in maths e.g. abstraction, and may also vary to reflect reading age and cognitive ability. • the word decimal is sometimes used in computing when actually the denary number system is meant. This can cause confusion with the decimal number system as it is taught in maths.