Review of international computing textbooks

December 2020

National Centre for Computing Education

Executive summary

The purpose of this international textbook review is to gather information on the nature of and use of textbooks in the teaching of computing to 5 to 16-year-olds in a number of countries that are high-performing and/or have a developed computing curriculum. In this review we examine how a selection of textbooks are used as a teaching tool and identify features of the textbooks from which we can learn when considering what makes a high-quality product.

The findings of this review are intended to support the development of an evidence base for good quality computing teaching in England and to inform textbook development for the English primary and secondary school market.

A number of countries were selected because of their potential contribution to textbook usage and textbook design: China, Estonia, Finland, Israel, Hong Kong, Japan, Kazakhstan, South Korea, New Zealand, and the United States of America. From these countries, ten textbooks were identified, reviewed, and in some instances, selected for deeper examination through research, recommendation, and classroom experience.

A variety of teaching approaches were evident within the textbooks including direct instruction, flipped learning, game-based, enquiry-based, kinaesthetic, mastery learning, and project-based. Overall, it was not possible to identify a significant difference in the pedagogies of non-western and western countries based on the textbooks selected.

In this report we identify features of high-quality textbooks which form the basis of recommendations. These features have been identified as contributing to their successful use in the classroom. Seven recommendations are made regarding promotion of the use of textbooks in the teaching of computing specifically. These are:

1. Computing textbooks should provide good lesson structure, including introductions, chunking, progression, assessment for learning, and assessment for teaching. They also need to be sensitive to individual student needs and so accommodate differentiation and extension.

2. Computing textbooks should be suitable for use by students with special educational needs, for example, using readability, white space, font style, highlighting keywords, colour-coding instructions, clear images, minimal clutter, and links to e-resources, including sound files.

3. Textbooks should have a consistent format, as this can reduce cognitive load when reading and thinking about the content.

4. Computing textbooks need to be clear about software and hardware dependencies.

5. Textbooks should adopt or promote sound classroom practice. What this means will depend on the approach taken in the curriculum for which the textbook is designed. The current English government promotes a knowledge-rich curriculum with pedagogic approaches that focus on subject content.

6. Guidance should be provided on the use of textbooks. For example, a textbook may not repeat content, but for reinforcement of concepts, teachers may be advised to revisit sections to support learning.

7. Textbooks should aim to cover curriculum content concepts in more than one way to solidify learning.

December 2020

This review was commissioned by the Department for Education in England and written by the National Centre for Computing Education.

National Centre for Computing Education

Funded by
Department for Education

Contents

Summary of stages

Stage 1: Identification of countries

In Stage 1, ten countries were agreed with the Department for Education for inclusion in this review. All the countries met the DfE's criteria of high performance[1] in the OECD's Programme for International Student Assessment (PISA) and/or ongoing curriculum development in computing. The countries are China, Estonia, Finland, Israel, Hong Kong, Japan, Kazakhstan, South Korea, New Zealand and the United States of America. Useful facts relating to their individual education systems are listed in Annex A for reference/context.

Stage 2: Identification of textbooks

In Stage 2, 49 textbooks were initially identified from the 10 countries from Stage 1. A shortlist of 10 was identified for deeper examination. These textbooks ensured a good spread of representation of the following factors:

- High performance in PISA
- Textbooks based on a complete computing curriculum
- Computer programming as well as computer science
- Meeting the needs of the youngest age range (learners under 7 years of age)
- Meeting the needs of the oldest age range (learners over 14 years of age)
- Non-English-language textbooks
- Authority-approved textbooks based on the state curriculum
- Authority-approved textbooks based on a qualification specification
- Books suitable for home purchase or student interest

The relative scoring of the textbooks against these criteria can be found in Annex B. The textbooks selected are given in Table 1.

[1] For the purposes of this review, "high performing" are those which score most highly in both mathematics and science as reported through the OECD's Programme for International Student Assessment (PISA) (2018). Data taken from the PISA 2018 report are combined to give a mathematics-cum-science rating. China (B-S-J-Z) is the best performing country, with 14 and the least-well performing country having 2. On this scale, the UK rating is 10.

Table 1: Textbooks selected for review, in alphabetical order of country

Country	**Textbook**
China	Primary Computing
	Hello Ruby
Japan	Hop, Step and Jump 308
	Information science 306
	New Information Science 310
Israel	Computer Science Concepts in Scratch
Kazakhstan	Informatics 10
New Zealand	Computer Science Unplugged
South Korea	Computers and Coding
USA	Big Fat Notebook

Stage 3: Analysis of textbooks

In Stage 3, the selected textbooks were scrutinised and an analysis undertaken to review their content and to identify how they are used by teachers and their students in schools, such as in lesson planning or delivery, testing students' knowledge, summative assessment, or homework.

This report identifies the pedagogical approaches promoted within the textbooks, and comments on their compatibility (or lack of compatibility) with a knowledge-rich approach to teaching. The school curriculum in England (but not necessarily in other countries) is defined as "knowledge-rich" and thus textbook features that support a focus on subject matter (as opposed to generic skills and competences) are highlighted. A detailed description of the format, content, and use of each textbook is given in Annex C.

Findings

In this section we detail the findings of the analysis of the ten textbooks under the following headings:

1. Summary of findings of textbooks from high-performing countries
2. Intended use of selected textbooks
3. Subject Matter
4. Approaches to teaching
5. Physical details

1. Summary of key findings of textbooks from high-performing countries

The high-performing countries were identified in Stage 1 as China, Japan, and South Korea.

In **Japan**, there are prescribed textbooks for computing. We selected three of these, Information 306, Hop Step Jump 308 and New Information 310, which have been in use for five or more years with a continual programme of review and revision. They are designed for 11- to 14-year-olds, and offer a subject-rich, whole-curriculum content presented in a systematic and prescribed subject order. They also include features such as:

- Coherent division of the subject content into chapters and subsections
- Comprehensive index
- Glossary of terms
- Short tests of knowledge and understanding
- Integrated skills development (software usage and programming)
- Highlighting of keywords
- Connected e-resources that reinforce, extend, or differentiate the learning experiences

In **China** there are no prescribed textbooks for computing, although there are for other subjects. The textbooks identified for China have yet to gain popularity or universal usage but are published in Chinese languages for the Chinese education and home markets. They are both for primary education, so it is not possible to compare directly with the Japanese textbooks. The textbook ‘Primary Computing’ is a substantial body of knowledge for computing for students from age 5 to 14 (8 textbooks, with 45 lessons, each of a duration of an hour or more). In contrast, Hello Ruby is a story-based introduction to computational thinking for 7- to 11-year-olds with 22 activities of about an hour in duration.

For **South Korea**, it was not possible to obtain a prescribed textbook so “Computers and Coding” was chosen for the review. Like the Chinese books, this textbook is intended for use by primary school aged children. It is written in an informal style and uses a lift-the-flap approach. It is a translation of a UK textbook, containing a small subset of the topics covered in the KS2 curriculum for England.

Pedagogical features of textbooks from elsewhere are described later in the report, but overall, it was not possible to identify a significant difference in the pedagogies of Asian and western countries based on the textbooks selected.

2. Intended use of selected textbooks

In this section we describe the different audiences and intended purposes of the selected textbooks. Some of the textbooks provide specific guidance on how they should be used. Primary Computing (China) has a Teacher's Guide (140 pages) per three textbooks; Informatics 10 (Kazakhstan) has a brief guide "How to use this book"; and Information Science 306 (Japan) has a 32-page teacher's guide called 'DIGEST'. Otherwise, the content of the textbook has been used to gather findings around how the textbooks are intended for use in the classroom.

Use to structure lessons

The content structure of the subject matter may have implications for the teaching approaches that may be used. Six of the selected textbooks facilitate lesson-by-lesson progress through knowledge-rich subject content:

- Hop, Step, Jump — Japan
- Kazakh Informatics 10 — Kazakhstan
- Primary Computing — China
- Information science — Japan
- New Information Science — Japan·
- Computer Science Concepts in Scratch — Israel

Specific examples of this are as follows:

- Informatics 10 (Kazakhstan) has a set topic structure. Each topic includes a description of aims and is structured with four sections: You will, Practice, Keep in mind, and Literacy. The latter section provides an assessment opportunity.
- The Computer Science Concepts in Scratch (Israel) is presented in a concept-by-concept sequence for the teaching of programming.
- The Primary Computing textbook (China) has the most lesson-orientated approach. Each double-page spread starts with "In this lesson" and a description for "Getting started". The majority of the layout provides instructions or information for the student, which the teacher can use to introduce, describe, and explain the subject content. There is always an "Activity" to promote student-focussed independent or collaborative learning. The "Think again" and "Extra challenge" blocks prompt the teacher to provide extension activities.

Use for differentiation

All teachers use differentiation[2] in their teaching to cater for the needs of students who may grasp concepts at a different pace to others. Providing differentiation is challenging for textbook writers. For any single topic, it means providing three or more versions of the same content but at different educational levels. In addition, some needs are very specific — for example, concerning the colour of the background for text, the size of text, or a Braille alternative.

The textbooks that are suitable for many students with special educational needs are those that ensure a degree of sensitivity to issues including readability, white space, font style (bat-and-ball a), highlighting keywords, colour-coding instructions, clear images, minimal clutter, links to e-resources (sound files), etc. Textbooks reviewed do meet some of these requirements. For example, in Hop, Step, Jump (Japan), the full texts are presented with a soft copy that can be edited. This enables the textbook to be used for students with specific needs, for example, the text can be enlarged for students with visual impairments.

Hop, Step, Jump (Japan) also exemplifies how to engage and enable the students with the least advanced computing skills of the group to succeed at their level whilst enabling the majority to progress and the students with the most advanced computing skills to be challenged. This is achieved through the structure of the book: Hop gives the outline of a topic, Step explains the content in depth, and Jump provides an extensive series of exercises for consolidation and challenge.

Use for reference/ by a teacher

One of the selected textbooks, Computer Science Unplugged (New Zealand), is intended to be used by the teacher, not the students. It provides activities for the teacher to organise for the students, with an explanation of how these relate to computer science knowledge and understanding.

Others that facilitate being used as a reference text, either by teachers for professional development or lesson planning and preparation, or by students for revision, include the following:

- Big Fat Notebook (USA) has an extensive index, demarcation of topics within the body of the textbook, and clear labelling of keywords.
- Hop, Step, Jump (Japan) has highly detailed contents pages, which aid the teacher and student to see the scope of each topic.
- Informatics 10 (Kazakhstan) has a references page to make connections with other publications.

Use for homework

A school using the Hop Step Jump textbook (Japan) states, “it is important to work at school and at home on a daily basis in order to improve the ability”. No other textbook explicitly mentions homework. Some textbooks are ideally suited to revision or study at home, for example, Big Fat Notebook (USA).

[2] differentiation’ means providing different experiences for different students to meet the individual students’ needs

Many of the textbooks have extension activities, differentiated questions, and further reading, that could be used for homework.

Use for assessment

Textbooks can be a useful resource in the assessment of computing, both as assessment for the student and as assessment of the student. Assessment for learning (AfL) can be implemented in several ways, including through peer assessment and self-assessment. The Primary Computing (China) illustrates the use of self-assessment, as the student is given heuristics or criteria by which to judge their own work.

Several textbooks provide the answers to the questions set, including the Big Fat Notebook (USA) — and give direct and immediate feedback to the student. The questions presented can act as templates for the teachers to create their questions for the class.

Another aspect of this form of assessment supports the teacher making judgements about the students' preparedness for examination and the probable grade that will be obtained. Only the Kazakh book had examination type questions associated with each chapter.

Use for storytelling

Hello Ruby (China) uses a classic story-based approach to introduce a scientific, knowledge-based subject to students. The book follows Ruby's journey, each chapter having a story and challenge that is solved through computational thinking. This approach can enthuse and engage the student. The storyline brings into focus the relevance of the learning activities.

3. Subject Matter and content

The subject matter of the textbooks is considered in relation to the Programme of Study for Computing as part of the National Curriculum for England.

Annex E gives the aims and high-level teaching topics from age 5-16. Computing is mandatory in maintained schools in England until age 16 (the end of key stage 4) but students can select optional qualifications beyond age 14, including the GCSE computer science (Annex F lists the subject criteria). Schools which are not maintained, such as academies and free schools, have freedom to design their own school curriculum, but are required to offer a broad and balanced curriculum to their pupils, with many using the computing curriculum as an exemplar.

The new National Curriculum for England, introduced in September 2014, is slimmer than the previous one and focuses on the essential knowledge that must be taught, allowing teachers to take greater control over the content and how it is taught. The Teach Computing Curriculum, produced by the DfE-funded National Centre for Computing Education (NCCE), includes a detailed breakdown of the curriculum through a ten-point taxonomy and learning objectives for each key stage, including for GCSE computer science. Annex G lists the units of work from the Teach Computing Curriculum, and Annex H, the coverage of the textbooks mapped against it.

This analysis shows that none of the textbooks cover the entirety of the Teach Computing Curriculum, which is to be expected as they are targeted at particular age groups. Two books which align most closely to the Programme of Study for England at age 14-16 are Kazakhstan's Informatics 10 book and the Big Fat Notebook from USA.

For the primary-focused textbooks, Hello Ruby (China) covers 8 of the 36 primary topics within the Teach Computing Curriculum and Computers and Coding covers 7.

Some textbooks, like Hello Ruby (China), are not comprehensive in that they do not cover a complete curriculum (see page 14). Textbook series like Primary Computing (China) are more likely to cover the whole curriculum. Single textbooks, like Big Fat Notebook (USA), can have the appearance of completeness, but care has to be taken that every element of the specified curriculum is covered.

Some textbooks link to other teaching matter. For example, the introduction to Informatics 10 (Kazakhstan) states that “a textbook is no longer the only source of information…”. This encourages independent learning, online research and highlights the connected nature of the computing subject matter.

4. Approaches to teaching

Whilst the selected textbooks have much in common with regards to their teaching approaches, there is variance in terms of their use in the classroom. For reference, a description of a range of teaching approaches is included in this report at Annex H, and a table indicating which teaching, and assessment, approaches have been identified within which selected textbooks is presented in Annex I. In this section we include a selection of different approaches to teaching and show how these could be used in conjunction with some of the selected textbooks. Not all textbooks are appropriate for each approach, but these examples illustrate the breadth of use that a textbook can provide.

This section includes a range of teaching strategies that may be adopted in the classroom, giving examples of textbooks that use that approach.

Direct instruction

The direct instruction approach involves explicit teaching of the content using lectures or demonstrations of the material to students. It may be supported by a well-structured textbook. A textbook designed to be used in the context of a largely direct instruction approach may prescribe the objectives and provides precision in the description of the expected activity, behaviour, or response required. It also provides in-built assessment that gives feedback on only the prescribed expectations.

Any textbook can be used with a direct instruction approach, and this is not necessarily mandated by the textbook contents. The only textbook solely devoted to teaching computer programming, Computer Science Concepts in Scratch (Israel), uses direct instruction with a lot of supporting explanation. For the younger age range, Primary Computing (China) provides a level of precision that may readily support a direct instruction approach.

Mastery Learning

Mastery learning maintains that students must achieve a level of mastery in prerequisite knowledge before moving forward to learn subsequent concepts. A mastery teaching approach takes mastery learning as the goal and incorporates repetition of activity and regular reinforcement of concepts.

Five of the textbooks, all designed for use for students over 14 have adopted some aspect of this approach. Informatics 10 (Kazakhstan), Hop, Step, Jump (Japan), Information science (Japan), New Information Science (Japan), and the Big Fat Notebook (USA) all provide a level of technically precise language.

Features of the selected textbooks that support a focus on content and concepts include highlighted keywords, a pupil-speak glossary, suggestions for unplugged activities,

suggestions for on-screen or online activities, open-ended questions for project- and problem-centred learning, questions with answers, examination-style questions, tips for learning including mnemonics, visual representations, an index page, a contents page, chapter and section abstracts, explicit statements of intention such as “in this section, you will learn…”, etc.

Independent learning

Some books lend themselves readily to independent learning. One textbook that is content-rich and could be used directly by students without being guided by a teacher is the Big Fat Notebook (USA). The writing is informal, using lined pages and a script-style font, and the text has a conversational style. However, the content is thorough and, providing it matched the examination specification, the textbook could serve as a revision guide and a reference source.

For a textbook to be used in an individualised setting, it must be accessible (language-wise), complete (covering the whole curriculum), and self-checking (including questions and tests that provide feedback to the student).

Two other textbooks, Computers and Coding (South Korea) and Hello Ruby (China), are written in an informal style as if to engage a student directly. However, they do not support an individual independently learning. Although these books represent the most informal approaches to teaching the computing curriculum, they both need a high level of teacher intervention.

Kinaesthetic

The kinaesthetic approaches to teaching including by-doing and by-being can enthuse, motivate, and engage students. The textbook that is the most hands-on is Computers and Coding (South Korea), using a ‘lift-the-flap’ approach. However, the kinaesthetic nature is limited to lifting flaps and the medium for learning is reading. The activities of Computer Science Unplugged (New Zealand) go much further by making the medium for learning a number of varied objects. Activities ranged from working with paper-based charts and graphs to the physical activity of running from place to place around the playground. No other textbook offers the same opportunity to take learning away from the paper or the screen.

Project-based learning

Within computing, there is an opportunity for project-based approaches to learning. Computers can be programmed to solve problems, and by setting challenges within the subject content area, learning can be facilitated. There are several examples in the selected textbooks.

For example, in Primary Computing (China), one project requires students to plan and implement a survey, and to gather and analyse data. The medium for learning is the learning content — spreadsheets, web pages, coding, etc. The project utilises the skills that are being developed. The textbook Computer Science Concepts in Scratch (Israel) uses mini-projects to focus the programming activity.

5. Physical details and other characteristics

1. Textbook format

Two of the textbooks have a hardback, low-page-number format (Hello Ruby has 114 pages and Computers and Coding has 18 pages); all the others have a glossy cover and soft-bound format. It is worth noting that cost is an important factor when considering the 'value' of educational resources, and soft-bound materials can be less-expensively produced. The format of the lift-the-flap Computers and Coding (South Korea) is necessarily hardback with internal card pages, adding to cost and reducing capacity for subject content.

Of the textbooks presenting a knowledge-rich content, the contents page varies from 68 sections or chapters across 174 pages to as few as 17 over 250. The richest contents page has one entry for every 2.5 pages of text. Some textbooks add a commentary to the contents in the form of a rationale. Hello Ruby (China) and CS Unplugged (New Zealand) provide a strong rationale for the subject content. A good overview or contents page enabled the reader to understand the systematic nature of the subject matter.

Hello Ruby (China), Computers and Coding (South Korea), Primary Computing (China), and Hop, Step, Jump (Japan) all have a glossy, colourful, and engaging appearance, which attract the reader.

2. Consistency of layout

All textbooks have a style. That style is unique to the actual textbook if it is a 'one-off', for example Computer Science Unplugged (New Zealand), whereas others are part of a series with a common format across the different aspects of computing or the age range of the series. The consistency of format reduces cognitive load when reading and thinking about the content.

Based on the textbooks in this survey, both western and Asian textbooks have a consistent style with regard to headings, subheadings, action boxes, and bullet and numbering styles. In particular, Information science 306 (Japan) , the Kazakhstan textbook and Primary computing (China) all have a consistent layout, page-by-page, topic-by-topic, and book-by-book.

The western textbooks tend to have a less cluttered layout with more white space, whereas the Asian textbooks have a higher amount of content per page.

Many Asian textbooks have images of people with speech bubbles drawing attention to learning points. Most textbooks use a lot of photographs to illustrate points. The textbooks for younger children more often use hand-drawn imagery.

3. Dependency on hardware and software

An aspect of some textbooks is the exemplification of, and often reliance upon, commercial products.

One textbook states "The learning content is explained in general theory and does not depend on a specific computer or software" but then requires an operating system of Microsoft Windows 7/8.1/10. Textbooks need to be clear about such software and hardware dependencies.

Many textbooks prescribe or utilise open-source software, for example, Informatics 10 (Kazakhstan) refers to LibreOffice Base, whilst others refer to the use of specific products, which are an additional cost to the school. Others refer to commercial products, and even use them in illustrations, but make similar use of open source or free-to-use software. Some textbooks are agnostic and make no reference to specific products, and the illustrations do not represent real software but a generic image, for example, as in the representation of a spreadsheet in Primary Computing (China).

An example: teaching algorithms

The teaching of algorithms is a topic that is central to computer science and the learning of programming. Table 2 shows how different textbooks approach the teaching of this topic, in order of intended age.

Table 2: Contrasting approaches to teaching algorithms

Country	Textbook	Age group	Teaching of algorithms
South Korea	Computers and Coding	5 - 11	One two-page spread, with a very simple flow chart.
China	Hello Ruby	7 - 11	Through a storyline and a set of associated activities
New Zealand	Computer Science Unplugged	9 - 16	Searching, sorting, minimum spanning tree routing and network algorithms are taught through practical, kinaesthetic activities
Japan	New Information Science 310	11 - 14	Flow charts are used to describe an algorithm, with examples of programming in both VBA and JavaScript
USA	Big Fat Notebook	11 - 16	Algorithms are taught in the context of computational thinking, alongside their implementation in a programming language

Features of high-quality computing textbooks

Following the review of individual computing textbooks, the following characteristics were determined to be of particular value when considering the features which contributed to a quality textbook. These aspects of good practice have been noted in some or all of the selected textbooks to a lesser or greater degree. Inclusion of these features within computing textbooks can directly increase their quality and value to both teacher and learner:

Fully comprehensive: Computing textbooks should cover all the age-appropriate subject content for the age range identified or the associated examination specification.

Supportive: Computing textbooks should support teachers' subject and pedagogic development through including suggested alternative approaches to teaching, reference to further subject matter resources that underpin or reinforce the content, and a description of the rationale for subject matter inclusion and sequence. For example, there are a range of supporting materials produced by the NCCE that focus on evidence-based pedagogical approaches for the teaching of computing[3], that may be referenced to support teachers' pedagogical content knowledge.

Coherent: Computing textbooks should have continuity and delineation of content that supports progression through the subject matter, so that the teacher can use them in a sequential order if they want to. Examination of the contents page should reveal the order and content of the subject matter, to help the teacher navigate the contents.

Relevant: Computing textbooks should avoid including material that does not directly contribute to the aims and objectives of the specified curriculum area or age range of students. This is particularly important with textbooks associated with a particular specification for examination or award.

Connected: Computing textbooks should reference other teaching and learning resources where possible. The concepts in computing will be better understood with relation to topical examples, the relevance of computing to society, and critical ways of thinking about issues. For example, there are now numerous case studies relating to the capabilities of artificial intelligence and machine learning in the context of computing applications.

Engaging: Computing textbooks should be attractive in presentation, activity, and content for both teacher and student.

Differentiated: Computing textbooks should provide tasks that are accessible and provide challenge. The text should be written with the reading ability of the target age group in mind, so that children can access the subject matter. Terminology should be clearly defined with reference where appropriate to other everyday meanings of the terms being introduced.

Reinforcing: Reinforcement of concepts is a feature of mastery learning. It is important that computing textbooks facilitate and encourage the reading and rereading necessary for learning. Practical programming exercises should include starter code which illustrates use of syntax and can be built on, and the use of 'copy code' activities should be avoided.

Consistent: Computing textbooks should use common formats throughout when presenting content e.g. for information, instructions, and questions. The use of consistent

[3] http://teachcomputing.org/pedagogy

colour labels, highlighting keywords, and the use of titles and subtitles all aid navigation and reduce cognitive load. Research suggests that including worked examples with sub-headings for stages of the process (known as sub-goal labelling) can support learning.

Accurate: Computing textbooks should be factually, typographically, and syntactically accurate. This has a bearing upon the perceived and actual value of any teaching resource. Inaccuracies, such as presenting a technical issue using an inappropriate metaphor, can bring doubt upon other parts of the work that might be totally correct.

Recommendations

In summary, from the reviews of the selected computing textbooks, and the identification of the features of a high-quality textbook, we have drawn out the following seven recommendations for selecting a textbook for a given computing curriculum.

1. Computing textbooks should provide good lesson structure, including introductions, chunking, progression, and assessment for learning. They also need to be sensitive to individual student needs and so accommodate differentiation and extension.
2. Computing textbooks should be based on recognised pedagogical approaches to the teaching of computing, for example, as highlighted by the NCCE at http://teachcomputing.org/pedagogy
3. Computing textbooks should be suitable for all, including students with special educational needs, for example, using readability, white space, font style, highlighting keywords, colour-coding instructions, clear images, minimal clutter, and links to e-resources, including sound files.
4. Textbooks should have a consistent format, as this can reduce cognitive load when reading and thinking about the content. For example, programming language and pseudocode extracts should use consistent indentation and use a monospaced font.
5. Computing textbooks need to be clear about software and hardware dependencies, and where possible ensure that only freely available software is used or recommended.
6. Guidance should be provided on the use of textbooks. This guidance should include the rationale for the pedagogical approaches being used, for example why in programming pupils are encouraged to read, understand and explain code before writing substantial amounts of code themselves.
7. Computing textbooks should aim to cover curriculum content concepts in more than one way to solidify learning. To illustrate this with examples: societal and ethical issues could be covered within a range of contexts and related to a range of different computer systems topics; design and testing principles could be reinforced by being related to a range of different topics; and core programming concepts need to be revisited over time, with gradually increasing complexity.

Annex A: Selected countries

Country facts – educational context

China

China's Compulsory Education Law stipulates that all children must complete nine years of compulsory education, starting at 6 or 7 years old. Children can be enrolled in kindergarten at the age of 2 or 3, but preschool education is not compulsory. Several languages and dialects exist across China; English is taught in some schools but is not used as the medium for education. Chinese education focuses on strictness and precision, which improves retention. Where computing education takes place, there is a focus on skills. The 'Digital China ICT4E' programme promotes technological solutions to close the gap between rural and urban schools with the emphasis on providing and using computing to support teaching and learning. University opportunities are self-funded by students; some university provision is provided by overseas institutions and sessions are taught in English. Textbooks in schools are prescribed centrally; however, there are no prescribed textbooks for computer science. Pupils have a school life expectancy of 14 years. There is no data on the spend on education. There is a 97% literacy rate. The population is 1.4 billion.

Israel

Israeli law requires all children from the age of 4 years to attend compulsory education (K-12). The primary language of education is Hebrew. All students must study English from grade 4. The 1995 introduction of computer science into the curriculum and the subsequent Science and Technology Excellence programme (STEP) in 2011 has created an environment where computing education continues to develop. Computer science is one of the subjects that high school students can select to study in depth but at no age is the subject compulsory. Textbooks do not play a significant role in the current provision – there are no prescribed textbooks for computing. Pupils have a school life expectancy of 16 years. The spend on education is 5.8% of the GDP. There is a 98% literacy rate. The population is 9 million.

Japan

Education in Japan is compulsory at elementary and lower secondary levels. Most students attend public (state-funded) schools through the lower secondary level, but private education is popular at the upper secondary and university levels. Education prior to elementary school is provided at kindergartens and day-care centres. English is taught in all schools from the age of 11 upward, with an emphasis on grammar rather than communication. English is not used as the medium for education. The computing curriculum is supported by set textbooks that are supplied through specific, government-approved publishers and are written to meet the detailed requirements of the national curriculum. In Japan, computer programming lessons are compulsory in elementary schools, but not all students study computing through to 16 years old and there are variations across the country. Pupils have a school life expectancy of 15 years. The spend on education is 3.5% of the GDP. There is no literacy rate data. The population is 126 million.

Kazakhstan

The education system in Kazakhstan is overseen by the Ministry of Education and administered at the local level. Schooling is mandatory for all students between the ages of 6 and 15. About 62% of children attend preschool. There are two pre university educational options — science-based (including informatics) and humanities-based — for students between the ages of 16 and 18. There are different types of school and each provides support for different abilities of students, for example, the BIL (Bilim Innovation Lyceum) schools and Nazarbayev Intellectual Schools. Access to those schools is granted by entrance examination or selection. The BIL schools provide support through a series of set textbooks for informatics. Lessons are taught in Russian and Kazakh, and from the age of 11, some lessons are taught in English, including informatics. Pupils have a school life expectancy of 15 years. The spend on education is 2.8% of the GDP. There is a 100% literacy rate. The population is 19 million.

South Korea

students start primary school at the age of 5. There are six years of primary school, three years of lower secondary school, and three years of upper secondary school; the latter is either academic or vocational. South Korea offers free half-day public (state-funded) preschools (called 'kindergartens') for students aged 3 and above. English is widely taught from 7 years of age but is not used as the medium for education. The South Korean Ministry of Education supports a software-centric computing education in the K–12 system (age 5-18), with an emphasis on software skills. All textbooks are prescribed centrally and supplied through specified publishers. There is no specific computing text. Pupils have a school life expectancy of 17 years. The spend on education is 5.3% of the GDP. There is a 98% literacy rate. The population is 52 million.

New Zealand

Education is free for students between the ages of 5 and 19 at state schools. Schooling is compulsory from age 6 to 16. Primary education is up to Year 9 (age 13) and secondary education is from Year 9 to Year 13 (13 to 17 years of age). English is the de facto official language spoken by 95.4% of the population, with Maori a de jure official language. Programming and computer science (a strand of Digital Technologies) is a qualification at age 16 and is taught in English. There are no prescribed textbooks. Pupils have a school life expectancy of 19 years. The spend on education is 6.4% of the GDP. The population is 5 million.

United States of America

Free access to education is available from the age of 5 to 18 (grades K–12) but varies between states. Education policy is decided and implemented at state level. English is the usual medium for education, but a minority of schools teach in Spanish, Chinese languages, and some indigenous languages. Promoted by the Computer Science Teachers' Association (CSTA)[4], where taught, computing in the USA includes programming, algorithms, devices, and data analysis. Only about a quarter of all high-school computer science classes in the USA use commercially published material on a weekly basis, and, of these, the most common textbooks relate to the scripting languages HTML and CSS. Any other textbook is used by less than 6% of computing classes (citation

[4] https://www.csteachers.org/

available). For younger students, storybooks have been recommended to teach computer science concepts. Pupils have a school life expectancy of 16 years. The spend on education is 5% of the GDP. There is no comparable literacy rate data. The population is 333 million.

Sources

- 2018 National Survey of Science and Mathematics Education www.horizon-research.com
- 2nd Survey of Schools: ICT in Education ec.europa.eu
- Digital China ICT4E digitalchina.blogg.lu.se/tag/ict4e
- National Center for Education Statistics (NCES) nces.ed.gov/programs/statereform/tab2_21.asp
- Storybooks That Teach Computer Science Concepts sites.google.com/view/literacy-in-k5cs
- Students, Computers and Learning: Making the Connection www.oecd-ilibrary.org/education
- The World Factbook www.cia.gov/library/publications/the-world-factbook
- Informal discussions with teacher colleagues working in the countries listed.

Annex B: Selection of Textbooks

Table 3: Textbooks selected for review, in rank order

Anglicised title	Country	Rank	PISA	Coding	Youngest	Oldest	Non-En	Curric	State	Qualif	Interest	Pts
Hop, Step, Jump	Japan	1	3	3	1		1	3	2			13
Kazakh Informatics	Kazakhstan	2		3		3		3	2	2		13
Primary Computing	China	3	3	2	1		1	3			2	12
Information science Society 306	Japan	4	3	2			1	3	2			11
New Information Science	Japan	5	3	2			1	3	2			11
Hello Ruby	China	6	3	3			1				2	9
CS Unplugged	New Zealand	7	1			2		2			4	9
Big Fat Notebook	USA	8	1	3		2					2	8
Computing and coding for cooking	South Korea	9	3	3	1		1					8
Computer Science Concepts in Scratch	Israel	10	0	3		1	1				1	6

Key
1 indicates it has a significant aspect 2 indicates that it has a good aspect 3 indicates that the contribution is a strength factor

Ranking factors	
PISA	place in the PISA tables
Coding	the degree to which the textbook covers the toping of programming
Youngest	does the textbook meet the needs of 5-year-olds?
Oldest	does the textbook meet the needs of 16-year-olds?
Non-En	does the textbook provide a non-English perspective?
Curric	does the textbook represent (or is it one of a series representing) a whole curriculum?
State	does the textbook receive government/authority endorsement?
Qualif	does the textbook meet the requirements of a specific qualification?
Interest	does the textbook have appealing characteristics that would engage students in addition to that of an interest in the subject?

Annex C: Textbook information

The textbooks

(1) Hop, Step and Jump 308 (Japan)

(2) Informatics 10 (Kazakhstan)

(3) Primary Computing (China)

(4) Information science 306 (Japan)

(5) New Information Science 310 (Japan)

(6) Hello Ruby (China)

(7) Computer Science Unplugged (New Zealand)

(8) Big Fat Notebook (USA)

(9) Computers and Coding (South Korea)

(10) Computer Science Concepts in Scratch (Israel)

ホップ、ステップ、ジャンプ (Japan)

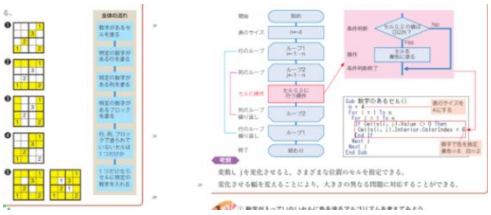
Hop, Step, Jump is a Japanese centrally authorised textbook for computing, covering a range of topics for a specific age group working towards level 2, formally referenced as 'Information Science 308 Information Science New Edition', published in 2017, in size B5 with 176 pages, ISBN: 978-4-407-20386-8. The authorisation process ensures accuracy of content and relevance to the curriculum. The presentation is of a high standard and of a consistent design. Most pages are mainly diagrams or illustrations. Where there is a lot of text, it is presented in chunks, labelled, and structured. The regular central review of the provision of textbooks ensures the up-to-date relevance and factual accuracy. That central control of textbooks ensures that the curriculum is comprehensively covered across the whole age range.

HOP covers the introduction to nine themes through 21 two-page practice exercises that establish a basic level of understanding of each topic. The nine topics are: Information and Computers, Networks, Algorithms and Programs, Modelling and Simulation, Databases, Problem Solving, Information Systems, Security and Development of Information Society, and Intellectual Property Rights. It ensures that students have the foundations of knowledge to best develop their understanding of the later material. This differentiation means that students of all academic abilities can achieve some success.

STEP consists of nine chapters, each of the different topics given above in depth. These sections reinforce the earlier learning and provide a content-rich experience for the student and a clear route for teaching. The textbook is connected; there is direct reference in the text to the supporting e-materials provided on a DVD.

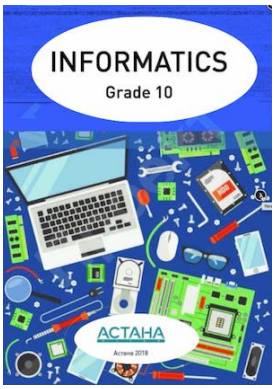
JUMP has nine hands-on extensive exercises to deepen and extend understanding of the themes. This provides challenge as well as supporting the teacher's knowledge and understanding of the curriculum aims and purpose. The textbook has a structure designed to enable the teacher to meet the needs of students at different academic levels using different styles of teaching. In particular, the textbook supports by-doing, direct instruction, flipped, mastery, objective-led, scaffolded, and teacher-centred approaches.

There is an integration of subject matter and the use of algorithms and programming to illustrate the subject matter. There is a high level of subject content, with the topics being covered in depth, reflecting a knowledge-rich approach to the subject, for example, considering patterns and logic, and then using programming and flow diagrams to illustrate the pattern-making. There are questions that are useful for tests and assessment activities. The two programming languages promoted are Scratch and Microsoft Excel VBA. The study notes are distributed as Microsoft Word documents and can be edited by the teacher or the students. Again, this supports the teacher in developing their approach to teaching the material.



There are connected resources including Study Notes for students, revision notes (All-Point Set), and a teacher's Guidance Book; and on DVD, a digital version of the textbook and Microsoft PowerPoint slides of the material.

информатика 10 (Kazakhstan)



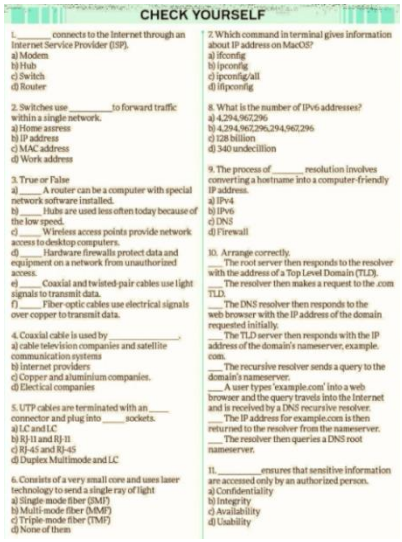
by Eldar Yerzhanov, Ismail Gesen, Altynbek Zhunussov, Nursultan Akhmetov, Yerzhan Shaniyev following 1. Оқу бағдарламасы Қазақстан Республикасы Үкіметінің 2012 жылғы 23 тамыздағы № 1080 қаулысымен бекітілген Орта білім берудің (бастауыш, негізгі орта, жалпы орта білім беру) мемлекеттік жалпыға міндетті стандартына сәйкес әзірленген.

Informatics 10 is an approved publication associated with a level 2* qualification offered by the Bilim Innovation Lyceum in accordance with the state compulsory standard of secondary education (approved by the Government of the Republic of Kazakhstan on August 23, 2012, No. 1080), comprehensively covering the topics of informatics[5]. It was published in 2018 and has 144 pages.

The textbook has clear graphics and many screenshots to illustrate the skills or knowledge being taught. It has an attractive and consistent layout and format: it is divided into eight chapters, each with between three and twelve subsections, reflecting the amount of content for each topic. The chapters are as follows:

1. Computer literacy [hardware/systems], including architecture, CPU, and motherboard
2. Database basics [number representation], including number systems, truth tables, logical operations, and encoding standards
3. Programming focussing on Python, including strings, functions, and dictionaries
4. Project work [software development], including software development trends, operating systems, and computer optimisation
5. Computer literacy [relational databases], including planning, creating, linking, forms, queries, SQL, and reports
6. Computer literacy [audio and image editing], including GIMP (raster graphics editor), OpenShot (video editor), and practical project work supporting learning by-doing
7. Computer literacy [web programming], including site structure, style, CSS, and JavaScript (variables, assignment, data type, operators, logic, and iteration)
8. Computer literacy [networking], including components, connectors, addressing, DNS, and ethics (the triad of confidentiality, integrity, and availability)

[5] Informatics is the study of natural and engineered computational systems, and is the name given to computer science in many countries in Europe, and Russia.



CHECK YOURSELF

1. _____ connects to the Internet through an Internet Service Provider (ISP).
a) Modem
b) Hub
c) Switch
d) Router

2. Switches use _____ to forward traffic within a single network.
a) Home assress
b) IP address
c) MAC address
d) Work address

3. True or False
a) _____ A router can be a computer with special network software installed.
b) _____ Hubs are used less often today because of the low speed.
c) _____ Wireless access points provide network access to desktop computers.
d) _____ Hardware firewalls protect data and equipment on a network from unauthorized access.
e) _____ Coaxial and twisted-pair cables use light signals to transmit data.
f) _____ Fiber-optic cables use electrical signals over copper to transmit data.

4. Coaxial cable is used by _____.
a) cable television companies and satellite communication systems
b) internet providers
c) Copper and aluminium companies.
d) Electical companies

5. UTP cables are terminated with an ____ connector and plug into _____ sockets.
a) LC and LC
b) RJ-11 and RJ-11
c) RJ-45 and RJ-45
d) Duplex Multimode and LC

6. Consists of a very small core and uses laser technology to send a single ray of light
a) Single-mode fiber (SMF)
b) Multi-mode fiber (MMF)
c) Triple-mode fiber (TMF)
d) None of them

7. Which command in terminal gives information about IP address on MacOS?
a) ifconfig
b) ipconfig
c) ipconfig/all
d) ifipconfig

8. What is the number of IPv6 addresses?
a) 4,294,967,296
b) 4,294,967,296,294,967,296
c) 128 billion
d) 340 undecillion

9. The process of _____ resolution involves converting a hostname into a computer-friendly IP address.
a) IPv4
b) IPv6
c) DNS
d) Firewall

10. Arrange correctly.
___ The root server then responds to the resolver with the address of a Top Level Domain (TLD).
___ The resolver then makes a request to the .com TLD.
___ The DNS resolver then responds to the web browser with the IP address of the domain requested initially.
___ The TLD server then responds with the IP address of the domain's nameserver, example.com.
___ The recursive resolver sends a query to the domain's nameserver.
___ A user types 'example.com' into a web browser and the query travels into the Internet and is received by a DNS recursive resolver.
___ The IP address for example.com is then returned to the resolver from the nameserver.
___ The resolver then queries a DNS root nameserver.

11. _____ ensures that sensitive information are accessed only by an authorized person.
a) Confidentiality
b) Integrity
c) Availability
d) Usability

A “Check yourself” activity is presented at the end of each chapter to support good classroom practice of self-assessment, peer assessment, and consolidation of knowledge. In turn, this supports a knowledge-rich approach to the subject. It consists of a page of closed, mixed-style questions focussing on the knowledge content and presented in the style of examination questions. There is a references section that lists 21 online and authoritative sources for the teacher and/or student and so supports the development of the teachers’ subject knowledge.

牛津国际青少年计算思维训练 (China)

by Alison Page et al., published by 清华大学

Primary Computing is a Chinese version of an international publication originally authored in the UK and published by Oxford University Press and now published by Tsinghua University Press. The series covers learning at entry level through to level 2* (age 5 to 14). Each textbook of the series covers a single academic year of study, and together, they form a comprehensive computing curriculum. This review considers Book 6 (students aged 10 to 12), published in 2020, with 104 pages.

This textbook supports a knowledge-rich approach to the teaching of computing and covers the following areas of the Teach Computing Curriculum: the internet, data logging, spreadsheets, web page design, and programming. It is divided into six consistently formatted chapters, each focussing on a different aspect of computing, which contain an online activity and class discussion to help students start thinking about the topic, followed by structured lessons to guide the student through activity-based learning grounded in good classroom practice. Each chapter has a “Check what you know” section with a test and activities to help measure students’ progress and inform the teacher of students’ misconceptions.

The chapters of Book 6 are:

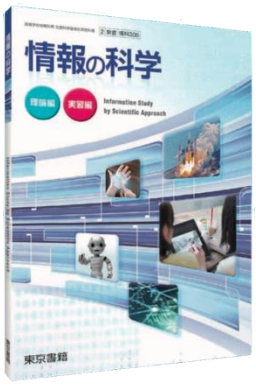
1. The nature of technology: Robots
2. Digital literacy: Make a web page
3. Computational thinking: Algorithms and programs
4. Programming: The Frog Maze
5. Multimedia: Our school survey
6. Numbers and data: Amir’s parcels

These cover aspects of control systems, web page design, algorithms, programming (Scratch), surveys, presentations, spreadsheets, and the social implications of technology.

The teachers’ guidance includes information about the underpinning philosophy, including spiral curriculum, activity-based learning, formative assessment, collaborative learning, and the use of meaningful (situated) tasks. The pages are characterised by high-quality images of real-life computing applications, illustrative graphics, and screengrabs of coding.

Each two-page spread has “Activity”, “Think again”, and “Extra challenge” items. At the end of each chapter is a “Check what you know” section with activities, a test, and self-evaluation to support continual assessment of the student.

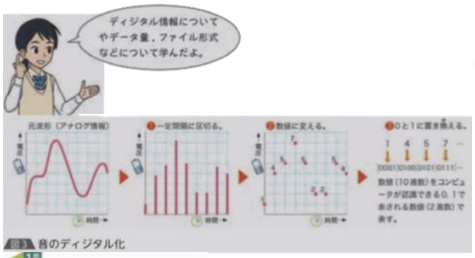
インフォメーション・サイエンス(Japan)



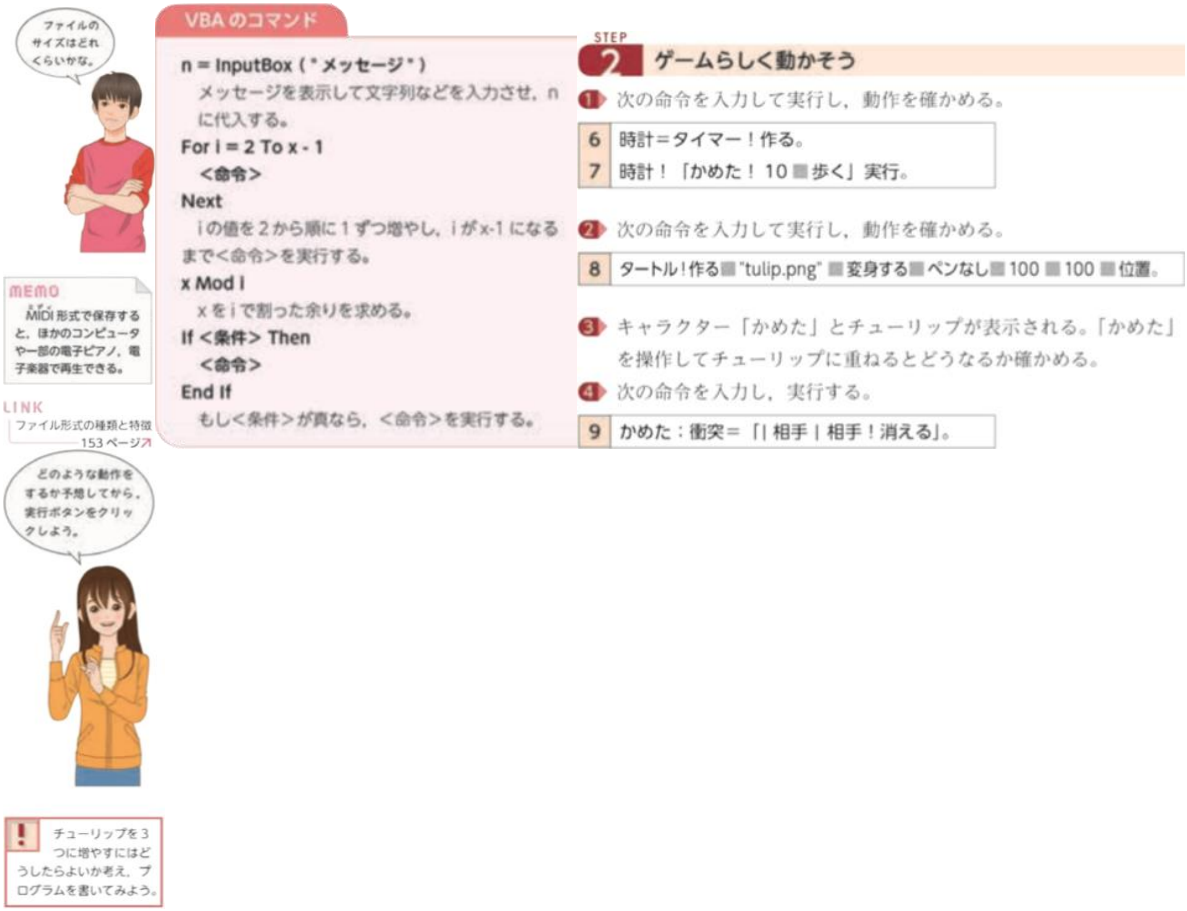
Information Science is a Japanese centrally authorised textbook for computing, covering a range of topics for a specific age group working towards level 2, formally referenced as ‘Information Science 306 Information Science New Edition’, published in 2017 by Tokyo Shoseki Co. Ltd. The authorisation process is overseen by the Ministry of Education, Culture, Sports, Science and Technology. 2019 saw the launch of the ‘GIGA School Concept’ to promote the enhancement of technology education. The regular central review of the provision of textbooks ensures that the curriculum is comprehensively covered across the whole age range and the resources remain up to date.

The presentation is of a high standard and of a consistent design. The pages have a high level of graphics, with drawn figures and speech bubbles drawing attention to important learning points. The information is presented in a way that supports didactic teaching, with step-by-step instructions suitable for mastery teaching approaches. There are links to online testing of the subject matter.

There are regular assessment points and answers published to provide further feedback to students about their progress. The publisher is supportive of teachers’ professional development through the accompanying online resources, which include direct reference to research and government policy, particularly that related to the “D-mark” of making strong connections between the paper-based and online resources.



Images (left to right): engaging and clear diagrammatic explanations; IT skills developed; regular test with answers provided.



Images (left to right): attention to important points; programming with embedded, step-by-step instructions; gender balance.

情報の新しい科学 (Japan)

表紙の説明

３次元コンピュータ・グラフィックスで用いられるポリゴンを使い、クジラとイルカを描きました。好奇心旺盛で、人なつっこいイルカは、情報社会に積極的に参画し、創造する態度を象徴しています。また、深海に潜っては、ときに海面から勢いよく飛び出すクジラは、「深い学び」と、自らの能力をいかんなく発揮する力強い姿を表現しています。

"Dolphins that sometimes jump out of the surface when diving into the deep sea express "deep learning" and a powerful figure that fully demonstrates their abilities."
It is the depth of understanding that is important, but expressed in open behaviour in practical exercises.

New Information Science 310 is a centrally authorised Japanese textbook of computing designed for 3rd year students (aged 14 to 15) and covers a range of topics. It was published by the Nihon Bunka Publishing Company in 2017, in size B5, with 194 pages. The authorisation to distribute textbooks ensures accuracy of content and relevance to the curriculum. The presentation is of a high standard and of a consistent design.

The curriculum topics covered form a sound sequence of learning for the specific age group. There is a consistency of layout, with each topic having side notes to explain the meaning of terms as they are introduced. The chapters are as follows:

- Chapter 1: Computer Processing and Representation of Information
- Chapter 2: Communication connected by Networks
- Chapter 3: Society Supported by Information Systems
- Chapter 4: Problem Solving and Automation of Processing Procedures
- Chapter 5: Modelling and Problem Solving
- Chapter 6: Information and Communication Networks and Problem Solving

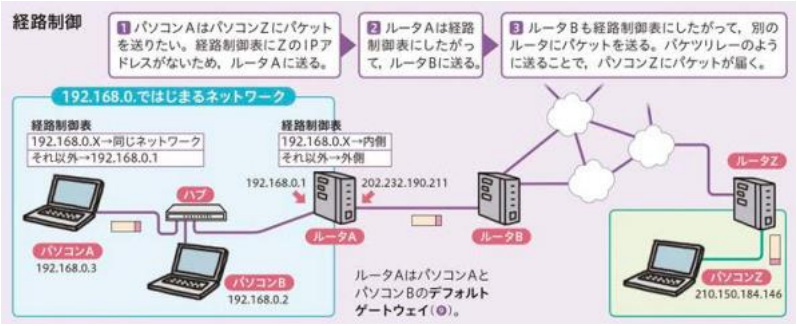
The textbook is characterised by a clear, professional, fact-only layout. This two-page spread represents how the computer works — CPU and logic circuit. Diagrams are not cluttered with irrelevant imagery but contain the essential facts represented in words and icons.

There are many confirmation questions and examples: reinforcement of the learning is promoted.

The examples use a range of programming languages.

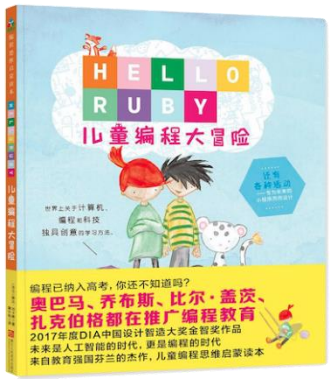
Problem-solving is presented as “academic skills”, promoting the idea that principles learned in computing can be applied elsewhere.

There is an introductory checklist that can be used by the teacher to judge the starting level of the students.



The textbook is connected with accompanying e-resources.

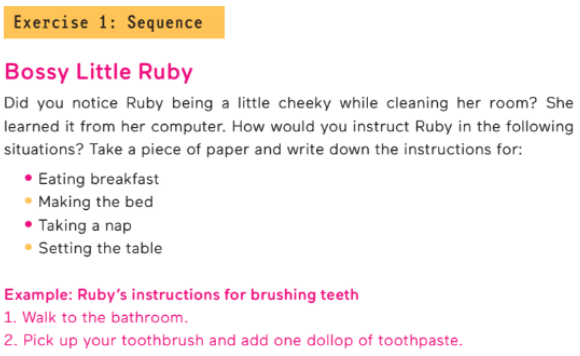
你好红宝石 (China)



by Linda Liukas

你好红宝石 Hello Ruby is a Chinese translation of the original storybook written in English? published by Zhejiang People's Fine Arts Publishing House, Zhejiang People's Beauty Publishing Company, and Beijing Baby Cube. With a teacher's support, students from 7 years old could be taught with this textbook; it has an age-appropriate context for students aged up to 11. It was published in 2017 and has 114 pages, ISBN: 978-7-5340-6004-5. The textbook introduces the characters and then tells the story of Ruby's adventures over ten chapters. The second half of the textbook is a sequence of 22 activities that introduce, explain, and reinforce computing concepts including sequence, decomposition, and pattern recognition; loops (iteration) and selection (what if/conditional); strings, numbers, Boolean operators, and data structures; creativity and computational thinking; functions and abstractions; and debugging, pair programming, and project development.

As you would expect from a textbook for students of this age range, there are many character images as well as illustrations and diagrams relating to the computing concepts. There are clear instructions for the parent/teacher and so it supports professional development. There is a teacher-speak glossary of terms. The approach is kinaesthetic, including learning by doing, engagement, and discourse. It is introductory in nature and does not cover topics to the depth necessary for the age group, but it is enthusing and ensures a positive attitude to the subject.


Exercise 1: Sequence

Bossy Little Ruby

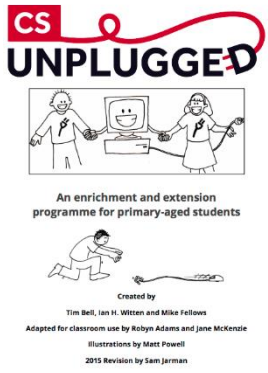
Did you notice Ruby being a little cheeky while cleaning her room? She learned it from her computer. How would you instruct Ruby in the following situations? Take a piece of paper and write down the instructions for:

- Eating breakfast
- Making the bed
- Taking a nap
- Setting the table

Example: Ruby's instructions for brushing teeth
1. Walk to the bathroom.
2. Pick up your toothbrush and add one dollop of toothpaste.

This textbook covers many of the topics expected of students in the age range of 8 to 12, specifically: technology and information technology around us, moving a robot, pictograms, robot algorithms, the internet, and repetition in shapes and games.

Computer Science Unplugged (New Zealand)

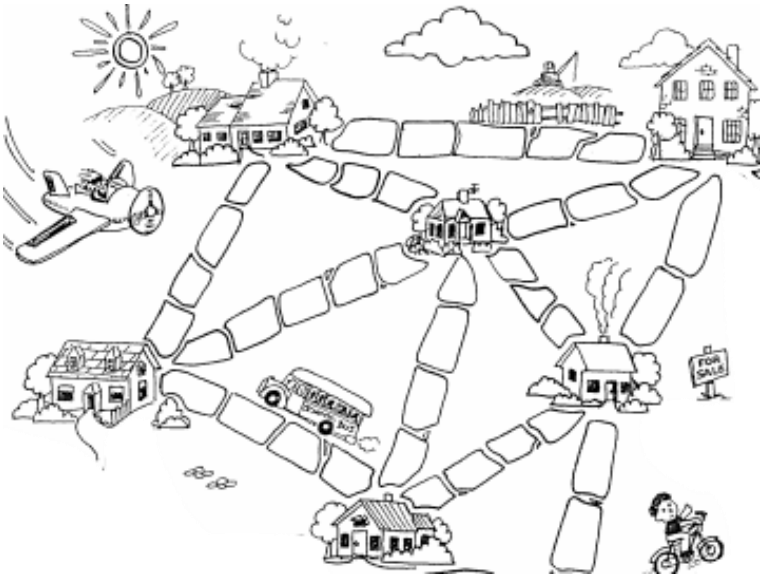


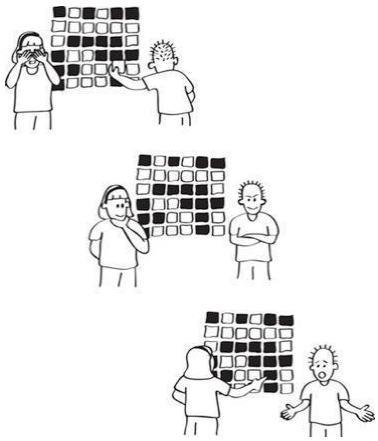
by Tim Bell, Ian H. Witten, and Mike Fellows

Computer Science Unplugged was revised in 2015 and has 236 pages: classic.csunplugged.org/books. This textbook contains enrichment and extension activities for the regular classroom, with background information explained simply. Answers to all the problems are provided, and each activity ends with “What’s it all about?”, an explanation of the relevance to the computing or mathematics curriculum.

Many of the activities are mathematically based, e.g. exploring binary numbers, mapping and graphs, patterns and sorting problems, and cryptography. Others develop a knowledge and understanding of how computers work. The students are actively involved in communication, problem-solving, creativity, and developing thinking skills in meaningful contexts.

Each of the 21 activities is unplugged, that is, not requiring a computer; they are generally kinaesthetic in nature, but all have an underpinning knowledge and understanding of subject content. The nature of the materials is such that some items can be used by the very young student and some can be used by the most experienced computing student. For example, ‘Muddy City’ is a colouring-in challenge for 5-year-olds in which they must connect the houses with the smallest number of slabs, yet the same sheet helps the more experienced computing students understand the implementation and workings of a minimum spanning tree algorithm. It provides a good example to illustrate abstraction.



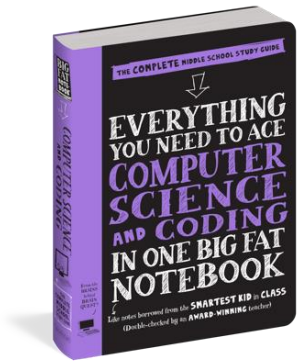


A magic trick (illustrated left) demonstrates the parity (check bit) process for confirming that packages of data are not corrupted in transmission. It is taking a complex concept and making it kinaesthetic and meaningful.

This is not a single textbook to cover the whole curriculum.

It is a textbook that enables the computing teacher to enthuse students in specific aspects of computing, including binary, image representation, compression, error detection, searching, sorting, finite state machines, encryption, and human-computer interaction.

Big Fat Notebook (USA)



by Grant Smith, published by Workman Publishing

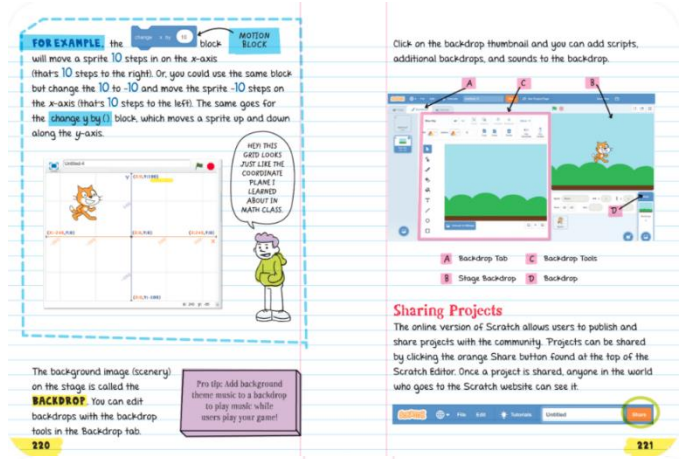
Everything You Need to Ace Computer Science and Coding in One Big Fat Notebook is an independent publication recommended for use by some schools and some states in the USA. The content is mapped against the CSTA Computer Science Standards Level 2 and is designed for students in the age range of 11 to 14. It was published in 2020, and has 576 pages, ISBN: 978-1-523-50277-6.

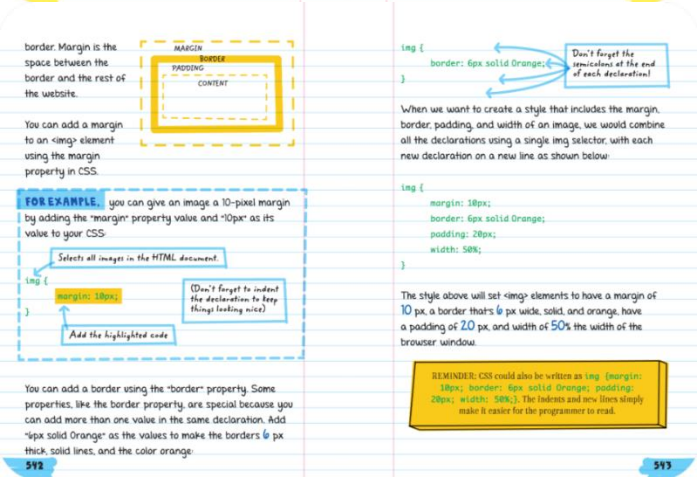
The textbook has 39 chapters divided between the following eight units:

1. Computing systems
2. Data and analysis
3. Software engineering
4. Algorithms and programming
5. Universal programming principles
6. Programming with Scratch
7. Programming in Python
8. Web development

These cover a complete computing curriculum of hardware, binary code, computational thinking, loops, events, procedures, Boolean expressions, cybersecurity, HTML, and CSS. Each chapter has a “Check your knowledge” page of questions — answers are provided and so it supports self-study. The pages are characterised by the notebook-style lines, a script-type font, occasional conversational prose, and a colourful layout. There is an extensive index.

This textbook is ideally suited to revision or study at home.





It covers the majority of topics of the NCCE's Teach Computing Curriculum for the age range of 13 to 16.

요리조리 열어 보는 컴퓨터와 코딩 (South Korea)

by Rosie Dickins, published by 어스본 코리아

“Computers and Coding” is a South Korean translation of a UK textbook, published by Earthborn South Korea. With a teacher’s support, students from the age of 7 could be taught with this textbook; it has an age-appropriate context for students aged up to 10. It was published in 2015, and has 18 pages, ISBN: 979-1-1868-7254-3.

The lift-the-flap textbook introduces, through text and graphics, nine aspects of computing:

1. What’s a computer?
2. What is coding?
3. What’s inside?
4. How computers think
5. Giving instructions
6. Computer language
7. Creating better code
8. Using the internet
9. History of computers

The pages are characterised by cartoon graphics, short phrase statements, computing symbols and representations, and flaps that reveal further text and graphics. The lift-the-flap approach has some kinaesthetic appeal, but the content is only portrayed in words and images. A student would need a teacher to help develop an understanding of the content. The lift-the-flap process is not a self-assessment tool. The activities are extended by links to online resources.

Topics of the Teach Computing Curriculum that are touched upon in this textbook include technology around us, moving a robot, events and actions, the internet, selection in physical computing, and programming. However, the textbook represents quite a narrow aspect of the key stage 2 computing curriculum. The amount of material would not support enough activity time to meet the needs of key stage 2 teachers. It would need to be supplemented by other materials.

מושגים למדעי המחשב בשריטה **(Israel)**

by Michal Armoni and Moti Ben-Ari, Weizmann Institute of Science

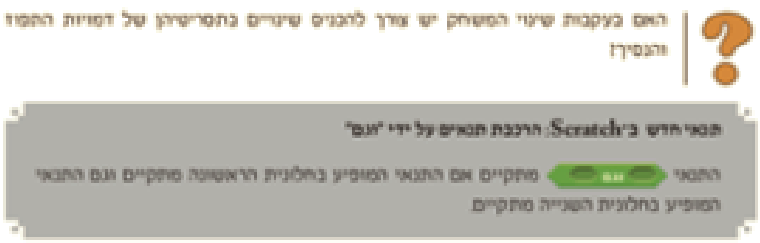
Computer Science Concepts in Scratch is written and designed by leading educationalists in Israel. It is primarily a tutor book in developing skills of programming in MIT Scratch written in Hebrew. English translations are available. It was published in 2010 and has 230 pages. The purpose of the textbook is to acquaint the student with concepts of computer science. It is ordered in terms of programming concepts and, although using a graphic development environment (MIT Scratch), it uses the proper language of computer science – for example using initialization, assigning variables, condition. It therefore can meet the needs of the older students.

The textbook has 11 chapters. The first 10 progress through programming constructs.

1 First Steps
2 Multiple Sprites
3 Short Scripts, Long Runs
4 Communications Between Sprites
5 On the Dance Floor—Repeated Run Again
6 Remembering Things—Variables
7 It Depends—Conditional Run
8 Numbers 1
9 Lists—Remembering Complex Information
10 Concurrent Run

Chapter 11 is interesting because it sets the challenges in the context of real-life computer issues. For example, the collision-avoidance radar reverses the usual challenge of detecting collisions to detecting the potential for collisions and avoiding them.

There is an introduction for students and one for teachers that fulfils a useful professional development role. In addition, the Looking Back section reflects on the rationale for teaching and learning computer programming. There is an extensive index.



The textbook is subject to a creative commons licence and so teachers can freely adopt and adapt the textbook to meet their needs and those of the students. There is consistent colour coding of sections and the text queries the students in a conversational manner – “Is

it necessary to make changes in the scripts of the orange characters following the change in the game?”.

It covers the area of the NCCE’s Teach Computing Curriculum for the age range of 11 to 16 related to computer programming.

Annex D: Computing National Curriculum[6]

Aims

The national curriculum for computing aims to ensure that all pupils:

- can understand and apply the fundamental principles and concepts of computer science, including abstraction, logic, algorithms and data representation
- can analyse problems in computational terms, and have repeated practical experience of writing computer programs in order to solve such problems
- can evaluate and apply information technology, including new or unfamiliar technologies, analytically to solve problems
- are responsible, competent, confident and creative users of information and communication technology.

Key stage 1 Pupils should be taught to:

- 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.

Key stage 2 Pupils should be taught to:

- 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

[6] https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study/national-curriculum-in-england-computing-programmes-of-study

- 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.

Key stage 3 Pupils should be taught to:

- design, use and evaluate computational abstractions that model the state and behaviour of real-world problems and physical systems
- understand several key algorithms that reflect computational thinking [for example, ones for sorting and searching]; use logical reasoning to compare the utility of alternative algorithms for the same problem
- use two or more programming languages, at least one of which is textual, to solve a variety of computational problems; make appropriate use of data structures [for example, lists, tables or arrays]; design and develop modular programs that use procedures or functions
- understand simple Boolean logic [for example, AND, OR and NOT] and some of its uses in circuits and programming; understand how numbers can be represented in binary, and be able to carry out simple operations on binary numbers [for example, binary addition, and conversion between binary and decimal]
- understand the hardware and software components that make up computer systems, and how they communicate with one another and with other systems
- understand how instructions are stored and executed within a computer system; understand how data of various types (including text, sounds and pictures) can be represented and manipulated digitally, in the form of binary digits
- undertake creative projects that involve selecting, using, and combining multiple applications, preferably across a range of devices, to achieve challenging goals, including collecting and analysing data and meeting the needs of known users
- create, re-use, revise and re-purpose digital artefacts for a given audience, with attention to trustworthiness, design and usability
- understand a range of ways to use technology safely, respectfully, responsibly and securely, including protecting their online identity and privacy; recognise inappropriate content, contact and conduct, and know how to report concerns.

Key stage 4

All pupils must have the opportunity to study aspects of information technology and computer science at sufficient depth to allow them to progress to higher levels of study or to a professional career.

All pupils should be taught to:

- develop their capability, creativity and knowledge in computer science, digital media and information technology
- develop and apply their analytic, problem-solving, design, and computational thinking skills
- understand how changes in technology affect safety, including new ways to protect their online privacy and identity, and how to identify and report a range of concerns.

Annex E: GCSE computer science subject content

Knowledge and understanding

GCSE specifications must require students to develop a knowledge and understanding of the fundamentals of computer science and programming including:

- standard algorithms, including binary search and merge sort
- following and writing algorithms to solve problems including:
    - sequence, selection and iteration
    - input, processing and output
- how particular programs and algorithms work
- the concept of data type, including integer, Boolean, real, character and string, and data structures, including records and one- and two-dimensional arrays
- representation of numbers in binary and hexadecimal; conversion between these and decimal; binary addition and shifts
- representation of text, sound, and graphics inside computers
- Boolean logic using AND, OR and NOT, combinations of these, and the application of logical operators in appropriate truth tables to solve problems
- the purpose and functionality of systems software, including the operating system and utility software
- characteristics of systems architectures, including:
    - CPU architecture, including Von Neumann and the role of the components of the CPU in the fetch-execute cycle
    - main and contemporary secondary storage and ways of storing data on devices including magnetic, optical and solid state
    - data capacity and calculation of data capacity requirements
    - hardware components and embedded systems
- networks and the importance of:
    - connectivity, both wired and wireless
    - types of network
    - common network topologies
    - network security
    - the concept of networking protocols, including Ethernet, Wi-Fi, TCP/IP, HTTP, HTTPS, FTP and email protocols
    - layers
- cyber security: forms of attack (based on technical weaknesses and behaviour),
- methods of identifying vulnerabilities, and ways to protect software systems (during design, creation, testing, and use)
- the ethical, legal and environmental impacts of digital technology on wider
- society, including issues of privacy and cyber security
- characteristics and purpose of different levels of programming language,
- including low-level language

Skills

GCSE specifications must require students to develop the following skills:

- take a systematic approach to problem solving including the use of decomposition and abstraction, and make use of conventions including pseudocode and flowcharts
- design, write, test and refine programs, using one or more high-level programming language with a textual program definition, either to a specification or to solve a problem
- use appropriate security techniques, including validation and authentication
- evaluate the fitness for purpose of algorithms in meeting requirements efficiently using logical reasoning and test data
- use abstraction effectively
    - to model selected aspects of the external world in a program
    - to appropriately structure programs into modular parts with clear, well-documented interfaces
- apply computing-related mathematics

Source: GCSE Subject Content for computer science

Annex F: The NCCE's Teach Computing Curriculum

Table 4: Units of work in the Teach Computing Curriculum by year of study

<table>
<tr><th colspan="2">Key Stage 1</th></tr>
<tr><td>Year 1
Computing systems and networks – Technology around us
Creating media – Digital painting
Creating media – Digital writing
Data and information – Grouping data
Programming A – Moving a robot
Programming B – Introduction to animation</td><td>Year 2
Computing systems and networks – IT around us
Creating media – Digital photography
Creating media – Making music
Data and information – Pictograms
Programming A – Robot algorithms
Programming B – An introduction to quizzes</td></tr>
<tr><th colspan="2">Key Stage 2</th></tr>
<tr><td>Year 3
Computing systems and networks – Connecting computers
Creating media – Animation
Creating media – Desktop publishing
Data and information – Branching databases
Programming A – Sequence in music
Programming B – Events and actions
Year 4
Computing systems and networks – The internet
Creating media – Audio editing
Creating media – Photo editing
Data and information – Data logging
Programming A – Repetition in shapes
Programming B – Repetition in games</td><td>Year 5
Computing systems and networks – Sharing information
Creating media – Vector drawing
Creating media – Video editing
Data and information – Flat-file databases
Programming A – Selection in physical computing
Programming B – Selection in quizzes
Year 6
Computing systems and networks – Communication
Creating media – 3D Modelling
Creating media – Web page creation
Data and information – Spreadsheets
Programming A – Variables in games
Programming B – Sensing</td></tr>
<tr><th>Key Stage 3</th><th>Key Stage 4</th></tr>
<tr><td>Year 7</td><td>GCSE
Algorithms part 1</td></tr>
</table>

Impact of technology – Collaborating online respectfully Modelling data – Spreadsheets Networks from semaphores to the Internet Programming essentials in Scratch – part I Programming essentials in Scratch – part II Using media – Gaining support for a cause **Year 8** Computing systems Developing for the web Introduction to Python programming Media – Vector graphics Mobile app development Representations – from clay to silicon **Year 9** Cybersecurity Data science Media – Animations Physical computing Python programming with sequences of data Representations – going audiovisual	Algorithms part 2 Computer systems Databases and SQL Data representations HTML Impacts of technology Networks Object-oriented programming Programming part 1 - Sequence Programming part 2 - Selection Programming part 3 - Iteration Programming part 4 - Subroutines Programming part 5 - Strings and lists Programming part 6 - Dictionaries and data files Security **Non-GCSE** IT and the world of work IT project management Media Online safety Spreadsheets

Source: http://teachcomputing.org/curriculum (NCCE November 2020)

Annex G: NCCE textbook coverage

Table 5: Textbooks mapped to the NCCE Teach Computing Curriculum content

NCCE Teach Computing Curriculum	Hop Step Jump	National 5 Kazakh Informatics	Computing Science Primary	Computing Information Science	New Information Science	Hello Ruby	CS Unplugged	Big Fat Notebook	Comp's & Coding	Year
Technology Around Us					√			√		
Grouping data										
Digital painting									1	5 - 6 yrs
Digital writing										
Moving a robot					√			√		
Introduction to animation										
Information technology Around Us					√			√		
Pictograms					√	√				
Digital photographs									2	6 - 7 yrs
Making music										
Robot algorithms					√	√		√		
Introduction to quizzes										
Connecting computers										
Branching databases										
Stop-frame animation									3	7 - 8 yrs
Desktop publishing										
Sequence in music										
Events and actions								√		
The internet			√		√			√		
Data logging			√							
Audio editing									4	8 - 9 yrs
Photo editing	√									
Repetition in shapes					√					
Repetition in games					√					
Sharing information										
Flat-file databases										
Video editing										
Vector drawing									5	9 - 10 yrs
Selection in physical computing								√		
Selection in quizzes										
Communication	√									
Spreadsheets	√		√							
Web page design			√							
3D modelling									6	10 - 11 yrs
Variables in games	√					√				
Sensing										
Collaborating online respectfully										
Media: gain respect for a cause										
Networks 1	√									
Spreadsheets			√						7	11 - 12 yrs
Programming 1	√		√	√				√		
Programming 2	√			√						
Media: vector graphics		√		√						
Computing systems	√	√		√	√					
Networks 2		√				√				
Representations 1		√		√					8	12 - 13 yrs
Programming 3		√			√					
Programming 4		√			√					
Media: animations, video		√		√						
Data science [infomatics]		√			√					
Cybersecurity	√	√			√	√			9	13 - 14 yrs
Representations 2	√	√					√			
Programming 5	√	√					√			
Programming 6	√	√					√			
Data representation	√	√	√		√	√	√	√		
Computer systems		√	√				√			
Networks		√	√		√		√	√	10	14 - 15 yrs
Security	√	√	√			√	√	√		
Impacts	√				√	√				
Algorithms		√	√				√	√		
Programming		√	√					√		
Databases & SQL		√	√						11	15 - 16 yrs
HTML		√	√					√		
Object Orientated Programming										

Annex H: Teaching approaches

Teaching approaches that could be used in conjunction with the selected textbooks

Table 6: Teaching approaches evidenced in the selected textbooks

By being	practical experience and highly contextualised learning
By doing	practical experience with highly contextualised learning
Differentiated	giving students of different aptitudes, academic abilities, and dispositions to learning different experiences, goals, or feedback
Direct instruction	being precise about the expected activity, behaviour, or response required, using teacher centred methods, and giving feedback on only those expectations
Flipped teaching	preparing the student through self-study and then providing an environment where students share their understanding and question where they do not understand
Game based	gamification of the learning process
Inquiry-based	giving students the opportunity and freedom to query all the assertions and knowledge being presented to foster a constructivist gained understanding
Kinaesthetic	learning through practical experience underpinned by constructionist principles
Mastery	learning promoted by repetition and 100% success experience
Objective led	teaching based on predetermined outcomes
Personalised	learning experiences tailored to the individual student's needs or interests
Problem-based	learning through meeting challenges, sometimes product focussed

Project-based	learning within a context that provides a range of experiences
Student-centred	see ‘Personalised’
Scaffolded	presenting new concepts making links to prior knowledge and understanding
Teacher-led	teacher-determined goals, activities, outcomes, and means of assessment

Annex I: Analysis of teaching approaches

The textbooks were analysed against the teaching approaches and opportunities for assessment identified in the Findings section.

Table 7: Textbooks mapped to potentially indicated teaching approaches

Textbook / Approach	by-being	by-doing	differentiated	direct instruction	flipped	game-based	enquiry-based	kinaesthetic	mastery	objective-led	personalised	problem-based	project-based	learner-centred	scaffolded	teacher-centred
Hop, Step, Jump 308		√		√	√				√	√					√	√
Informatics 10		√		√	√				√	√					√	√
Primary Computing									√	√					√	√
Information science 306				√	√				√	√					√	√
New Information Science 310				√	√				√	√					√	√
Hello Ruby	√	√		√			√	√			√	√		√		√
CSUnplugged		√		√		√	√	√				√				√
Big Fat Notebook				√	√				√	√	√					√
Computers and coding								√			√			√		√
Computer Science Concepts in Scratch		√		√	√				√	√		√			√	√

Table 8: Assessment approaches

Textbook / Assessment	self/peer/AfL	teaching
Hop, Step, Jump 308	√	√
Informatics 10		√
Primary Computing		√
Information science 306	√	√
New Information Science 310	√	√
Hello Ruby		
CSUnplugged		
Big Fat Notebook	√	√
Computers and coding		
Computer Science Concepts in Scratch		

The opportunities for assessment by the student for the student's benefit (tests with answers or feedback from peers) are used in some textbooks. Others provide tests that are teacher-marked and provide data to inform planning and teacher assessment.