<u>Inspector subject training guidance:</u> primary computing

<u>The purpose of this document</u>

This document has been created for training and supporting inspectors to conduct subject deep dives in schools. The training guidance provides a structure to explain variation in subject-level quality of education. It should be used in conjunction with handbooks for section 5, section 8 inspections of good and outstanding schools, and section 8 no formal designation (subject-specific) inspections.

<u>Points to consider when examining the evidence:</u>

School leaders may not be able and should not be expected to articulate their intent **as it is outlined** in this document or to provide documents which neatly provide the evidence for these focus areas. Inspectors should always investigate claims that issues affecting quality of subject education are outside of the school's control. It should be evident that the issue has been identified prior to the inspection and that the school has taken steps to mitigate the ill effects. For example, in the case of text books, it should be clear that leaders have previously identified the issue and raised it with senior leadership, investigated funding, identified texts they would prefer, identified the specific weaknesses of the current text and taken specific action to mitigate against those weaknesses.

<u>The structure of this training guidance:</u>

The six focus areas
These provide a structure to explain reasons for the quality of subject education as identified by inspection activities. Inspection activities are likely to be an iterative process as inspectors consider the evidence. Under each focus area there is one row and two columns.

Column 1: This is an outline of potentially stronger practice in the area each question explores. It also provides likely responses and other evidence inspectors may encounter and gives explicit guidance on how to interpret these responses.	**Column 2:** This is an outline of weaker practice in the area each question explores. It also provides likely responses and other evidence inspectors may encounter and gives explicit guidance on how to interpret these responses.
Inspector Questions: These are organising questions which, together, cover the relevant points inspectors need to investigate under each focus area. These questions serve as headings and are not designed to be asked of school leaders. There are examples of useful **school-friendly** questions inspectors might ask of people or the evidence to explain reasons for the quality of subject education. This is **not** a comprehensive list of questions which may be asked. Inspectors should use their own judgement but will find the school-friendly question suggestions useful.	

Six focus areas

1. The school's understanding of progress in computing and how that informs its approach to the curriculum
2. The extent to which teaching supports the goals of the computing curriculum
3. The effectiveness of assessment in computing
4. The extent to which there is a climate of high subject expectations where a love of the subject can flourish
5. The quality of systems and support for staff development
6. The extent to which whole-school policies affect the capacity for effective computing education

Inspectors are likely to use the following sources of evidence in making their judgements.

They will generally use:

- interviews with subject lead (if there is one) and/or the appropriate senior leader
- curriculum plans
- pupils' work
- discussions with pupils
- interviews with teachers
- lesson visits, including conversation with teachers, if possible.

Where appropriate, inspectors may use:

- the school's own records of lesson visits in the subject
- the resources available for teaching the subject (incl. school library, ICT facilities)
- the school's assessment policy
- assessment instruments, including mark schemes if there are any (not internal data)
- how the school provides pupils with feedback on their work
- how the school promotes the value of the subject, including via enrichment activities
- forms of support for inexperienced, non-specialist or struggling staff
- any support provided for the subject lead
- performance management's role in improving subject provision
- details of the timetable and staffing (including details of experience and qualifications of staff)
- school policies on teaching, assessment, homework, behaviour
- documents analysing strengths and weaknesses of the subject and any associated improvement plans.

Focus area 1: The school's understanding of progress in computing and how that informs its approach to the curriculum

<table>
<tr><th>Outline of potentially stronger practice in terms of intent, implementation and impact
NB: answers will take many forms. Below are common findings to look out for</th><th>Outline of weaker practice in terms of intent, implementation and impact
NB: answers will take many forms. Below are common findings to look out for</th></tr>
<tr><td colspan="2">Inspector question 1:
Scope: How does the school understand what it means 'to get better' (progression) in the subject, and does the school give meaningful attention to all categories of knowledge in computing? Is the scope commensurate with that outlined in the national curriculum?

School-friendly questions:
What is the purpose of computing within the curriculum?
What form does the computing curriculum take? (discrete, topic based, cross-curricular?)
What types of knowledge are valued within computing? Do pupils know these?
How do pupils develop theoretical and practical knowledge?</td></tr>
<tr><td>Computing is a rigorous academic subject which has links with the disciplines of mathematics, science and engineering. A high-quality computing education equips pupils to use computational thinking to become creators of digital technologies, digital artefacts and computing knowledge.

Across all key stages the national curriculum (NC) progression can be categorised in the following categories:
Computer science (CS)
Information technology (IT)
Digital literacy (DL)
Computer science

Computer science is the core of the computing curriculum and covers principles such as data representation, algorithms, data structures and programming. This provides the foundation knowledge required to understand and interpret other areas of the curriculum.</td><td>The planned curriculum samples aspects of the NC but does not match it in scope.

The planned curriculum does not match the scope of the NC. It may sample elements of CS, IT or DL but does not cover these in their entirety. Computing may be used to support other curriculum areas through topic work; however, there is no clear appreciation of how pupils 'get better' at computing through the different categories of knowledge.

The school relies on an external scheme of work and has not considered if it offers an ambitious computing curriculum.

The school makes use of an external scheme of work for computing. This may exist individually or be part or a wider 'packaged' curriculum. Little thought has been given to the scope of the curriculum offered.

The subject's curriculum plans do not cover the ambition of all aspects of the national curriculum.</td></tr>
</table>

Information technology Information technology provides a context for the use of computers within society. Within IT there is a focus on knowledge of how computers are used within different sectors and describes the methods to create digital artefacts such as videos, animations or 3D models. **Digital literacy** Digital literacy is the knowledge and ability to use technology confidently, competently and in a safe way. It covers wide-ranging knowledge from how to operate devices at a mechanical level, searching and selecting information and how to use digital devices safely and responsibly.	Curriculum plans lack detail or ambition around progression in one or more categories of knowledge (CS, IT, DL). Plans may neglect one or more categories of knowledge. The curriculum may focus on the personal interests of leaders or teachers without consideration of what it is important for pupils to learn.
Inspector question 2: **Scope and components**: Does the school ensure wide-ranging and expanding knowledge of computer science? **School-friendly questions:** ▪ Within the computing curriculum, what knowledge of computer science do pupils learn? ▪ How does pupils' knowledge of algorithms increase in complexity? ▪ How does the curriculum enable pupils to 'get better' at programming? ▪ Do pupils understand the role of data in computer systems?	
Pupils develop a strong schema of algorithms. Curriculum plans are well sequenced to develop a strong schema of algorithms over time. Pupils will know that algorithms can exist separately from computer programs. There is an expectation that key algorithmic vocabulary is learned and used. **There is a clear understanding of how pupils 'get better' at programming.** There are goals defined for the programming knowledge pupils are expected to develop as a result of the learned curriculum. These may be described in terms of what children are able to do; however, the underpinning knowledge is easily identifiable. The three core constructs of sequence, selection and repetition are secured through repeated and increasingly complex encounters. Leaders are clear what this looks like	**Curriculum planning does not identify the important knowledge that pupils will need to secure.** Curriculum planning identifies activities that pupils will engage in but does not identify the important knowledge to be taught. **Curriculum plans require pupils to engage in complex composite tasks without securing component knowledge.** Pupils may be expected to write complex programs such as complete games, but curriculum plans do not enable pupils to secure component knowledge of programming or the programming language used. In this scenario, pupils will find these tasks artificially difficult or require disproportionate scaffolding specific to the problem, which does not aid in developing transferrable knowledge. Scaffolding may only take the form of making cosmetic or limited changes to pre-written programs and not progress beyond this.

<table>
<tr>
<td>across years. Plans demonstrate how pupils learn the programming language they are expected to use.

Pupils are taught the components useful in developing knowledge of data.

Pupils are taught how data can be represented in different ways, such as numbers, words, images, audio and video. They learn that how data flows through a computer system as input, process and output. Pupils will develop knowledge of how data can be grouped and ordered.</td>
<td></td>
</tr>
<tr>
<td colspan="2">Inspector question 3:
Scope and components: Does the school ensure wide-ranging and expanding knowledge of information technology?

School friendly questions:
▪ What knowledge is it important for pupils to develop to create (spreadsheets/presentations/videos/images/animations)?
▪ How is pupils’ knowledge of how computers are used developed over time?</td>
</tr>
<tr>
<td>The curriculum to prepare pupils for creative projects identifies necessary component knowledge towards these composites.

Teachers should ensure that pupils learn the knowledge they need to be confident using applications in creative projects, including those that analyse data or manipulate digital artefacts. Subject leaders and teachers may describe pupils’ development of ‘skill’ within and across these applications, but in curriculum plans the capacity to use these applications should be underpinned through the identification of component knowledge. Examples of this knowledge might include:

▪ commonly used formulae within spreadsheet software (for example, sum and average) in addition to frequently used methods such as sorting, filtering and charting data

▪ in image editing: knowledge of bitmap/vector images, layers and colour blending. Knowledge of how to use common tools such as crop, rotate, scale. Knowledge of how to paint freehand and combine shapes

▪ in presentation software: knowledge of slide composition and rule of thirds. Appropriate font styles and sizes for purpose.</td>
<td>The creation of digital artefacts is limited to a single piece of software which pupils use in every year.

Pupils develop knowledge of a single application for the creation of digital artefacts. Plans will see pupils using this application for uses beyond its intended scope (for example, using presentation software for word processing, image editing or desktop publishing). This limits the knowledge pupils can secure.

Curriculum plans lack detail of the knowledge pupils will be developing through IT projects.

Curriculum plans for IT specify projects and expected outcomes. They do not highlight the knowledge pupils will gain through pursuing project work. Detail may focus on the qualitative aspects of the product, such as aesthetics that do not match curriculum goals.</td>
</tr>
</table>

<table>
<tr><td>Knowledge of image copyright. How to edit master slides, format images and apply animation.

The curriculum is designed to develop knowledge of computing contexts over time.

Pupils will develop knowledge of different computing contexts over time. Pupils will develop knowledge of how computers are used at home, in school and in businesses, and also global contexts such as the world wide web.</td><td></td></tr>
<tr><td colspan="2">Inspector question 4:
Scope, components, rigour: Does the school ensure wide-ranging and expanding knowledge of digital literacy?

School friendly questions:
▪ How do you ensure that pupils learn to use digital devices safely, efficiently and responsibly?
▪ How does a pupil’s knowledge of e-safety build over time? What would be different between KS1/KS2?</td></tr>
<tr><td>The subject identifies digital literacy knowledge required to use computing devices effectively and plans a curriculum to develop automaticity.

Curriculum plans identify essential declarative and procedural knowledge necessary to use hardware and software effectively. Teachers will plan for pupils to develop automaticity in this knowledge to reduce cognitive load when using devices for complex tasks. This knowledge is likely to include:
▪ physical use of input and output devices such as the mouse and keyboard (including typing)
▪ shortcuts (for example, ctrl/cmd + x/c/v, ctrl/cmd + s)
▪ file management (saving, opening and navigation)
▪ use of network and learning platform(s).

Pupils develop a secure knowledge of how to search for and select information.</td><td>Teachers believe children to be ‘digital natives’ and do not identify and plan for progression within digital literacy.

Plans will not show opportunities for pupils to secure knowledge in the operation of computing devices, effective use of the WWW or how to stay safe or behave responsibly online. Subject leaders may believe that this knowledge has already been secured through pupils’ interaction with digital devices beyond school.

The curriculum for digital literacy lacks cohesion.

The curriculum for digital literacy is a series of unconnected experiences that do not relate to each other (for example, internet safety day). There is no evidence that knowledge of behaving responsible online is developed through the curriculum.

The subject has no plan for progression within digital literacy.

Subject leaders will describe how pupils are exposed to digital literacy through the curriculum. When describing progression within digital literacy, this may be described as a skill that develops over time without concrete knowledge or application of methods to support being digitally literate.</td></tr>
</table>

<table>
<tr>
<td>Pupils develop knowledge of how search engines work and how information is ranked. Component knowledge of how to search for information effectively includes:
▪ appropriate key word choice
▪ use of advanced search techniques such as the use of quotes, Boolean operations (AND/OR), filters (such as – to remove unwanted terms) and ranges
▪ how to filter and sort results.
Plans will identify how pupils learn to discern if digital content is reliable. This will be built upon knowledge of features that are likely to make content less or more reliable and how the reliability of digital content can be verified against other sources.</td>
<td></td>
</tr>
<tr>
<td colspan="2">Inspector question 5:
Components, Sequencing: Does school planning consider component content and its sequencing to build knowledge over time and create ‘readiness’ for future learning? Is ‘ambition’ or ‘challenge’ considered in terms of identification of the knowledge, built over time, that will allow ambitious curriculum end points?
a. within the lesson sequence
b. within the topic
c. within the year or phase

School-friendly questions:
▪ Show me a curriculum example where specific computing content is sequenced to enable pupils to be ‘ready’ for something more complex.
▪ Show me how your curriculum prepares pupils for a particular topic through the knowledge that came before it.</td>
</tr>
<tr>
<td>The school plans for an integrated curriculum that combines all three aspects of the subject.
Despite separately identifiable categories of knowledge within the subject, the school recognises the links between these aspects and plans a curriculum where sequences are entwined to build knowledge in CS, IT and DL synoptically.</td>
<td>Curriculum plans show sequencing inherited from external schemes, textbooks or specifications without critical evaluation of sequence appropriateness.
The curriculum is planned as a series of disconnected units or lessons focusing on a narrow aspect of the subject where pupils may not revisit knowledge for long periods of time, and rarely see the interconnectedness of the subject.
Plans may show units that sample aspects of the computing curriculum. Plans may label units such as ‘spreadsheets’, ‘Scratch’, ‘Kodu’ or ‘animation’.</td>
</tr>
</table>

<table>
<tr><td></td><td>In discussions with leaders, they are unable to describe how these units build upon prior learning and develop knowledge through the CS, IT or DL aspects of the subject. Pupils may encounter concepts only once across a key stage.</td></tr>
<tr><td colspan="2">Inspector question 6:
Memory: Do teachers identify, emphasise and repeat crucial content so that pupils know more and remember more (i.e. make progress)?

School-friendly questions:
Show me which bits of your curriculum (like concepts, ideas, vocabulary, etc.) are crucial to re-visit so that they are remembered.
How do you identify and ensure pupils remember the most crucial content covered?
How do you as a school go about agreeing which specific knowledge (ideas, concepts, vocabulary, etc.) pupils absolutely need to know within each topic you teach?</td></tr>
<tr><td>Teachers plan for automaticity in commonly used procedures and methods.
Teachers plan to reinforce and develop strong schema of common procedures and methods. Pupils will be able to use these methods with automaticity, reducing cognitive load and allowing them to engage successfully in complex tasks. Examples:
secure knowledge of programming syntax, constructs and patterns (variables, repetition, conditional statements, data structures, input and output)
commonly used processes in the creation of digital artefacts (font manipulation, image editing, cropping, import/export).</td><td>Pupils lack automaticity when tackling complex tasks.
Pupils will be unable to easily draw upon prior knowledge of commonly used procedures and methods. This will lead to increased cognitive load and hinder opportunities for success in complex tasks.
Plans do not show enough time allocated for practice.
Plans may show limited time for pupils to practise methods and procedures.</td></tr>
<tr><td colspan="2">Inspector question 7:
Rigour: How does planning ensure the interplay of different categories of knowledge, thus ensuring pupils are given the capacity to consider subject-specific questions for themselves?

School-friendly questions:
Tell me how the different types of knowledge that you teach combine together in each topic?
You have stated that you wish for pupils to learn to develop ‘computational thinking’ or to think more like a subject expert. How have you planned curriculum content to ensure they have learned what they need to attain this goal?</td></tr>
<tr><td>The curriculum is constructed to deliberately develop computational thinking.</td><td>The subject leader believes problem solving to be a generic skill.</td></tr>
</table>

Computational thinking is a domain-specific approach to solving problems and applies to a range of methods used within the discipline by experts. Definitions of computational thinking are still emerging so teachers may describe this in different terms; however, common to all definitions are the concepts of decomposition and abstraction: Decomposition is the process of breaking down larger problems into smaller solvable sub-problems. Abstraction is the process of removing unimportant details when thinking about a problem or a system. The curriculum should allow pupils to develop their capacity to think computationally through repeated practice with carefully selected known and unknown problems sharing the same 'deep structure properties' (see glossary) This will be underpinned by secure pre-requisite component knowledge. Pupils should have the opportunity to implement solutions through practical programming. **Pupils get to consider where computing sits within the wider world, its history and its impact.** Subjects will plan for opportunities for pupils to think about the disciplinary qualities of computing. They will consider the history of computing, of which there is a rich British history. Pupils will be encouraged to debate the nature of the subject and its impact in their community and in the wider world, both in existing and emerging technologies such as AI, robotics, data science and digital privacy. This debate will be supported by secure subject knowledge so that pupils can engage in argument about the substance of the subject rather than surface level features.	The curriculum lacks appreciation that problem solving is domain specific. Leaders cannot describe how through their planning they develop pupils' capacity to solve problems or to think computationally. This may be described further to be a way of thinking that is applicable to other subjects. Where suggestions are offered, this is likely to be through pupils' exposure to repeated unconnected problems. **Computational thinking may be addressed in a way that does not make connections with programming.** Pupils do not see computational thinking as being connected to solving problems using computations. Exercises may only be limited to day-to-day processes such as describing processes for making jam sandwiches.

Inspector question 8:
SEND: How do you ensure those pupils who find it most difficult to learn computing (e.g. some pupils with specific SEND) are given the best chance to keep up?

School-friendly questions:

- Which pupils in this class are finding the subject most difficult? Why do they find the subject hard?

▪ Which bits of content are absolutely key that all pupils, including those with SEND, need to take away from this specific unit?	
The school identifies insecure key pre-requisite knowledge required before teaching new content and plans to secure it. The school plans to secure missing pre-requisite knowledge. This is particularly important for the hierarchical aspects of the subject, such as programming, to avoid dysfluency and avoid knowledge gaps. **Teachers are clear on curriculum content that is essential for all pupils to secure.** Through plans or discussions with the subject leader, there is a clear understanding of which aspects of the computing curriculum are essential for all pupils to learn, including those who find it most difficult to learn computing.	**The school provides a computing curriculum provision for groups of pupils that lacks ambition and does not prepare them for future study.** The school offers a different curriculum provision for groups of pupils, such as those with SEND, that is not commensurate with the NC. It lacks rigour in subject content and has little accountability. In lessons the experience of these pupils looks significantly different to their peers. **Pupils are removed from their computing lessons for additional teaching in other subjects.** Pupils who may already find computing difficult are removed from computing lessons for catch-up or intervention work in other subjects such as English or mathematics. This increases the difficulty of the subject for pupils as they lose out on opportunities to progress within the subject.
Inspector question 9: **Early education:** How effectively does the early education curriculum (EYFS and KS1) prepare pupils for their learning in key stage 2? **School-friendly questions:** ▪ How is the knowledge that children learn in the early years built on as they move through KS1 and KS2? ▪ How have you adapted the computing curriculum and teaching in the early years, considering any gaps demonstrated in KS2?	
Early education is used to establish foundation knowledge in computing. In computer science, pupils can develop knowledge of sequence by describing events using time connectives such as first, next and then. Pupils gain knowledge of how to use devices such as robots that are used more widely in KS1, reducing cognitive load when they come to use them to solve more complex problems. In information technology, they gain knowledge of choosing suitable devices and programs which are early steps in using computers purposefully and productively.	**No link is made between what pupils learn about technology in the early years and what pupils learn about computing in KS1.**

In digital literacy, they explore online safety or dilemmas through listening to stories or develop knowledge of how to use a mouse or keyboard when engaging in phonics activities.	

Focus area 2: The extent to which teaching supports the goals of the computing curriculum
Investigation will be necessary to ascertain if any issues with teaching in lessons visited illustrate a systemic issue with teaching in the school.

Outline of potentially stronger practice in terms of intent, implementation and impact NB: answers will take many forms. Below are common findings to look out for	**Outline of weaker practice in terms of intent, implementation and impact** NB: answers will take many forms. Below are common findings to look out for
Inspector question 10: Is the rationale for the teaching approaches chosen primarily to achieve the curriculum intent? What is the rationale for the teaching approaches chosen for sequences of lessons? **School-friendly questions:** ▪ Tell me a bit about the teaching approaches you have chosen in this sequence of lessons – what made them suitable for the content that you were teaching? ▪ Can you give me some examples of how the content that pupils study shapes the activity you have chosen to teach it?	
In teacher exposition, expert thinking will be modelled. Teachers will model the thinking normally undertaken by experts in computing. This will be through explicit explanation and demonstration of strategies used with opportunities for pupils to practice under supervision This may take various forms such as: ▪ selecting suitable variable names ▪ how and why a problem is decomposed into specific component parts ▪ using applications efficiently and thinking about the purpose and audience of media they create	**There is an over-reliance on online courses to teach programming.** Teachers will use online courses to teach programming. This may be with limited teacher instruction and be largely self-directed. Use of these platforms cannot offer flexibility of sequencing components and often develops only a surface level understanding of programming constructs. This can indicate a lack of teacher subject knowledge. **Pupils are not taught the most efficient or effective methods.** Pupils are taught inefficient or ineffective strategies to make up for a lack of secure component knowledge. Teachers provide 'workarounds' that may take longer, saturate working memory, increase later problem complexity or

▪ use of specific search terms, file names or directory structures. **Teachers make use of subject-appropriate scaffolding strategies when teaching programming.** Specific teaching strategies are used to reduce cognitive load when tackling problems and to enable high levels of success. Teachers may describe the use of the following methods: ▪ Parson’s problems: Complete prewritten programs or algorithms, where lines or blocks are mixed up and need correct arrangement. These problems may include distractors. ▪ Subgoal labelling: Small textual labels are provided for component steps of a problem-solving process in worked examples. ▪ PRIMM methods where pupils take a staged approach to programming through states of: predict, run, investigate, modify and make.	prevent schema development as pupils do not see the interconnectedness of concepts. Examples of this in practice include: ▪ using long sequences of commands or blocks in programming which could be more easily achieved using repetition ▪ using multiple variables to store values when a data structure would be most appropriate ▪ applying the same changes to multiple slides instead of using master slides ▪ applying formatting to individual paragraphs rather than changing style definitions ▪ using caps lock instead of shift for single capital letters ▪ writing a whole sentence or question to search the world wide web with a search engine.
Inspector question 11: What approaches do teachers use to ensure that key content is remembered long term? How do teachers ensure that pupils remember that which they have been taught? **School-friendly questions:** ▪ Show me some examples of where teaching activities were specifically chosen for pupils to remember things long term. ▪ Tell me a bit about how the approaches your school uses ensure that pupils remember what they’ve been taught. ▪ Can you show me some examples of approaches your school uses to support pupils in remembering what’s on the computing curriculum over time?	
Teachers create opportunities for pupils to recall and reuse frequently required prior knowledge. Teachers identify the knowledge that is frequently reused within categories and across categories. This is recalled and reused using strategies such as low-stakes quizzing, drills or short focused tasks requiring the use of essential procedures. Teachers will use opportunities to develop automaticity in the use of component	**There is an emphasis on making learning fun and engaging at the expense of securing curriculum goals.** Teachers will teach subject content through creative but often distracting activities. Children will spend a disproportionate amount of time thinking about the activity rather than the subject content. It is the nature of the activity they are likely to remember long term rather than the subject knowledge. This may manifest itself when teachers plan for pupils to play games incorporating or demonstrating subject content.

knowledge for more hierarchical aspects of the curriculum such as programming. **Teachers identify opportunities to reinforce knowledge across aspects of the subject when teaching new material.** When teaching new material, teachers will reinforce prior knowledge across computer science, IT and digital literacy. This might be to draw upon programming knowledge to implement a newly learned concept. Pupils may be asked to combine images and text in DTP software, and use an image they've created or edited in graphics software.	**Pupils will show an over-reliance on adult support when tackling challenging problems.** This may take the form of individual intervention with pupils, teaching of large parts of the task or providing detailed scaffolding materials that cover much of the process. This indicates a lack of secure component knowledge before tackling composite tasks. Although most frequently seen in programming, this may manifest itself in any area of the curriculum.

Focus area 3: The effectiveness of assessment in computing

Outline of potentially stronger practice in terms of intent, implementation and impact NB: answers will take many forms. Below are common findings to look out for	**Outline of weaker practice in terms of intent, implementation and impact** NB: answers will take many forms. Below are common findings to look out for
Inspector question 12: How does the school assess pupils' progress in learning computing? Does formative assessment identify the curriculum components pupils have not remembered or have forgotten? **School-friendly questions:** ▪ Tell me a bit about what your school thinks is the most effective way to assess pupils' progress in computing. ▪ Which bits of the curriculum do you prioritise when you construct assessments for pupils? Why do you prioritise these?	
The curriculum is the progression model and teachers will look to ensure that computer science, IT and digital literacy is are secured long term. Teachers assess prior knowledge regularly through formalised assessment opportunities and through teaching. This is used to identify lack of component security which is remedied before pupils are expected to perform composite tasks.	**Teachers only focus on the assessment of composite tasks.** All assessment will be focused on the completion of composite tasks such as writing computer programs, producing digital artefacts or presenting to an audience. Teachers will not identify deficiencies in component knowledge. **The subject assesses skills using vague descriptors or an abstracted mastery model.**

	The subject has an assessment model that focuses on computing skills. It uses vague descriptors or 'can-do' statements and does not focus on the substance of what children know or are able to do through progression in the curriculum. **Pupils' work is not readily available or difficult to access for teachers.** Systems and processes used within the subject make it difficult for teachers to see pupils' work on a regular basis or to see how this has improved over time. This creates a barrier to effective assessment.

Focus area 4: The extent to which there is a climate of high expectations where a love of the subject can flourish
This focus may well help explain the success of some schools, but a lack of evidence for 'climate where a love of the subject could flourish' could NOT reasonably be deployed to explain weakness given the challenge of identifying this during inspection. Curriculum expectations are covered above. Here, the question refers to how the school ensures that children put their best effort into their work.

Outline of potentially stronger practice in terms of intent, implementation and impact NB: answers will take many forms. Below are common findings to look out for	**Outline of weaker practice in terms of intent, implementation and impact** NB: answers will take many forms. Below are common findings to look out for
Inspector question 13: Does the school ensure that there are high expectations of children and that they respond to these expectations? **School-friendly questions:** ▪ How do you ensure pupils rise to your high expectations? For example, what actions do you take to ensure all pupils put their best effort into written work? ▪ Tell me how pupils with special educational needs might fare studying your computing curriculum?	
There is a clear inclusive approach to the subject where there are high standards for all. Teachers believe that computing is for all pupils, and all groups of pupils will be taught computing. With this, teachers have high	**There is a belief that not everyone can be successful within computing.** Teachers and pupils hold the belief that computing is only for certain groups of people such as boys, HPA or those that are 'good at maths'.

expectations for all groups to make progression through the curriculum. **Teachers make clear what progression looks like in the subject and pupils can see their success.** The subject will have clear criteria or objectives so pupils can see if they are achieving success and are motivated to do well as a result. Pupils recognise that they know, remember and can do more as a result of the teaching they receive. Tests and assessments support this.	
Inspector question 14: How does the school enrich the curriculum beyond classroom learning? **School-friendly questions:** ▪ Tell me a bit about what happens in the computing curriculum outside of the classroom. ▪ Are there any computing-specific experiences linked to the curriculum that take place outside of computing lessons? How do they link to the curriculum sequence? ▪ In what ways do pupils, who are very keen on your subject, get to share their enthusiasm?	
Teachers offer a range of clubs which enhance the curriculum. Teachers offer clubs which add greater depth or breadth to the curriculum. Clubs will offer opportunities to engage pupils in competition both locally and nationally. **The school plans opportunities for computing trips either within subject or as part of larger STEM opportunities.** The school will provide opportunities for pupils to engage in computing trips that broaden and reinforce their computing learning. Trips take opportunities to consider current computing, future employment or study and consider the past through the history of computing.	**The only enrichment opportunities are those that offer supervision when using computers.** The school offers supervised 'clubs'. Pupils use this time to play games or watch videos on computers. This works to erode the profile of the subject within the school.

Focus area 5: The quality of systems and support for staff development

<table>
<tr><th>Outline of potentially stronger practice in terms of intent, implementation and impact
NB: answers will take many forms. Below are common findings to look out for</th><th>Outline of weaker practice in terms of intent, implementation and impact
NB: answers will take many forms. Below are common findings to look out for</th></tr>
<tr><td colspan="2">Inspector question 15:
What do the strengths and weaknesses already identified indicate about the school’s capacity to function effectively?

School-friendly questions:
Tell me a bit about how inexperienced or struggling staff are supported.
How are teachers of computing enabled to develop their subject knowledge?
How easy is it to incorporate specialist software or hardware into curriculum planning?</td></tr>
<tr><td>The subject leader is an expert in computing.

The subject leader has secure subject knowledge through qualification, previous employment or subsequent training and demonstrates a passion for the subject. The subject leader is engaging with ongoing with opportunities to further develop subject knowledge and pedagogy.

Teachers are well supported in developing subject and pedagogical subject knowledge.

The subject leader demonstrates a clear plan in developing and supporting teachers’ subject and pedagogical subject knowledge. Where there is a lack expertise or capacity within the school, they leverage LA or MAT resources or make use of high-quality external support such as the government-funded National Centre for Computing Education and through computing at school.</td><td>The subject does not have the resources it needs to deliver an ambitious computing curriculum.

The subject may not have the hardware or software resources necessary to deliver an effective computing curriculum. This may be through a lack of resources or through policies which restrict the hardware or software that can be used.</td></tr>
<tr><td colspan="2">Inspector question 16:
How does the school go about the process of computing curriculum construction, debate and renewal?

School-friendly questions:
Tell me a bit about the process for curriculum is designed in your school. Is tweaking possible? If so, who decides on the changes?
Tell me about opportunities that staff have to feed back to you about whether the sequence of the computing curriculum is working.</td></tr>
</table>

There is a collaborative approach to curriculum construction and evaluation. All teachers of computing should have a role in curriculum development. Teachers are able to identify problems in curriculum design and contribute to solutions.	**The subject uses a cross-MAT/LA or bought in scheme and carries out no further evaluation or development.** Teachers are highly reliant on an externally obtained curriculum and there is no evidence of development or debate to critique or improve the provided plans. **The curriculum is built around outdated schemes of work.** Little review or renewal of curriculum plans has taken place recently. Indicators might include reference to ICT and little coverage of all categories of knowledge within plans.
Inspector question 17: How are all staff in the school encouraged to develop their subject knowledge and knowledge of how to teach that subject knowledge? **School-friendly questions:** ▪ Do you think that staff in your school are aware of their subject knowledge areas of expertise and areas for development? ▪ What opportunities do staff have to grow in knowledge and confidence about the topics that they teach? ▪ What place does subject knowledge have within the school's programme for CPD in computing? ▪ Are there any barriers that are preventing staff to develop their subject knowledge and teaching expertise?	
There is ongoing development of subject knowledge and pedagogical subject knowledge. The design and teaching of an ambitious computing curriculum is predicated on expansive and secure subject knowledge of computing. There should be clear evidence of ongoing development of subject knowledge for all teachers. The subject lead plays a pivotal role in fostering and providing opportunities for this to happen. **The subject lead promotes and supports engagement with computing subject networks.** There are strong expansive subject networks within computing, many offering local hubs across the country. They offer opportunities to engage in curriculum thinking and support development of subject knowledge. Subject leads will encourage staff to engage and	**Staff do not receive opportunities or support to develop their subject knowledge.** The school does not plan for opportunities for staff to develop their computing subject knowledge. The subject lead lacks the expertise or capacity to provide training to teachers and external training opportunities are not identified and promoted.

contribute to these communities and will have strong involvement with any local hubs that may operate. **Teachers develop expertise from other schools.** Teachers work with other local schools to develop subject knowledge. This may involve work with local secondary schools or, where the school is part of a multi-academy trust, other schools in the same trust. **Robust curriculum planning helps to mitigate subject knowledge gaps.** Curriculum planning is explicit in what children need to learn as part of the computing curriculum. Explanations, definitions and appropriate pedagogies are described in detail to support teachers who have gaps in their subject knowledge.	

Focus area 6: The extent to which whole school policies affect the capacity for effective computing education
This section is crucial to identify where the quality of education is influenced by the activities of the school and where the quality of education provided can be attributed to senior leadership.

Outline of potentially stronger practice in terms of intent, implementation and impact NB: answers will take many forms. Below are common findings to look out for	**Outline of weaker practice in terms of intent, implementation and impact** NB: answers will take many forms. Below are common findings to look out for
Inspector question 18: To what extent do whole-school policies reflect statutory obligations for computing education in the school? **School-friendly questions:** ▪ Is computing clearly taught in the curriculum from the start of KS1 to the end of KS2?	
School leaders and subject leaders take seriously the requirement that computing must be taught to all pupils at every key stage (National	School leaders and subject leaders do not take seriously the requirement that computing must be taught to all pupils at every key stage (National Curriculum

Curriculum Framework Document, December 2014, 3.6). Senior leaders ensure that there is sufficiently regular time within the timetabled curriculum that pupils can make progress by 'knowing more and remembering more' of the taught computing curriculum.	Framework Document, December 2014, 3.6). Computing is not taught to all pupils in every key stage, and pupils therefore cannot 'know more and remember more'.
Inspector question 19: What are the priorities for discussions at line management meetings between subject leaders and SLT?	
Line management meetings are used to discuss the substance of the subject Computing curriculum development and review are discussed as part of line management meetings. There is a focus on the challenges of teaching the computing curriculum and how these might be overcome.	**Line management meetings primarily focus on assessment data and pupil intervention.**
Inspector question 20: What criteria are used to decide on how curriculum time is allocated to computing?	
Enough time is allocated to teach the computing curriculum. Leaders will allocate enough time to ensure that the computing curriculum can be taught. Where the computing curriculum is integrated as part of topic work, at least some elements of the curriculum should be taught discretely.	**Insufficient time is given to ensure an ambitious curriculum is offered and for pupils to develop secure subject knowledge within it.** There is no dedicated time allocated for computing on the timetable. Where schools incorporate computing in topic work, it is not easily identifiable within plans. Less curriculum time is offered to computing during years of national exams.
Inspector question 21: How do school-wide policies, such as marking or CPD, support the subject's needs? **School-friendly questions:** ▪ Tell me a bit about how big-picture decisions in school affect computing. ▪ Is there anything about whole-school policies that limits or holds back the computing curriculum and assessment of it?	
School-wide approaches consider subject-specific nuances within computing. Senior leaders are sensitive to how school-wide policies can impact on computing. Senior leaders work with the subject lead to ensure an ambitious curriculum can be taught and assessed without compromising subject-specific practices.	**Computing curriculum time is used for other purposes.** The school sacrifices computing curriculum time to support other whole-school needs. This might be for intervention in other subject areas such as English.

Glossary

Term	Description
Algorithm	A defined process or set of rules to be followed in problem-solving activities. Typically implemented within a programming language to be executed by a computer.
Automaticity	Ability to recall and deploy (facts, concepts, and methods) with **accuracy and speed** and without using conscious memory; frees the working memory for higher-order processes that require holding a line of thought.
Components	The building blocks of knowledge or sub-skills that a pupil needs to understand, store and recall from long-term memory in order to be successful in a complex task. See Automaticity.
Composites	The more complex knowledge which can be acquired or more complex tasks which can be undertaken when prior knowledge components are secure in a pupil's memory.
Cumulative dysfluency	Educational failure caused when pupils do not have enough opportunities to recall knowledge to gain automaticity with the use of that knowledge. Over time this may cause many gaps in pupils' knowledge which prevent or limit pupils' acquisition of more complex knowledge.
Boolean logic	A form of algebra with values that equate to either true or false usually denoted by 1 or 0. This forms the basis of basic circuits within computer systems and is used to evaluate conditions within software.
Deep structure	The different ways a principle can be applied that transcend specific examples. When a principle is first learned, it is used inflexibly as the learner will tie that knowledge to the particulars of the context in which the principle has been learned (the 'surface structure'). As a learner gains expertise through familiarity with the principle and its applications, their knowledge is no longer organised around surface forms, but rather around deep structure. This means that experts can see how the deep structure applies to specific examples and that is an important goal of education.
Digital artefact	A digital object created using software and stored electronically. Examples include: programs, video files, sound files, presentations and animations.
Disciplinary knowledge	Methods and conceptual frameworks used by specialists in a given subject, e.g. knowledge of history or geography as a discipline.

Hierarchical subjects	Subjects where content has a clear hierarchical structure and there is often less debate about content choices than for cumulative subjects. This is because there are core components of knowledge that you must know in order to be able to progress within the subject. It would be hard to argue for a mathematics curriculum that didn't include algebra or place value.
Long-term memory	Where knowledge is stored in integrated schema, ready for connecting to and for use without taking up working memory. See schema.
Progression model	The planned curriculum path from the pupil's current state of competence to the school's intended manifestation of expertise.
Repetition (programming)	A programming construct allowing steps within a program or algorithm to be repeated. Programming language syntax commonly implements repetition using key words such as REPEAT, FOR and WHILE. This is one of the three key programming constructs pupils should develop knowledge of.
Schema/schemata (plural)	A mental structure of preconceived ideas that organises categories of information and the connections between them.
Selection (programming)	A programming construct allowing choices or decisions to be made in programs or algorithms. Most commonly these take the forms of IF style statements. This is one of the three key programming constructs pupils should develop knowledge of.
Sequence (programming)	A programming construct defining the order that the steps in a program or algorithm will be carried out. This is one of the three key programming constructs pupils should develop knowledge of across programming languages.
Substantive knowledge	Subject knowledge, often that carries considerable weight in a given subject domain, such as significant concepts.
Understanding	We are using the cognitivist model in which understanding describes pupils' interconnected knowledge e.g. of facts, concepts and procedures in maths. Understanding describes a certain schematic pattern of knowledge and is not qualitatively different from knowledge. Mental schemata can be viewed as network node diagrams, where nodes represent knowledge (facts, concepts, processes, features) and arcs the relationships between them. Understanding in this model is a function of the quantity of appropriate nodes and the quantity of appropriate arcs - more knowledge, and more connections between them leads to more understanding. A knowledge schema can always be developed further, and this is synonymous with deepening understanding. In this sense, a curriculum plan articulates the degree of understanding intended. In everyday life, the question 'do you understand?' invites a binary yes/no response. This implies that understanding is something that is finite and can be possessed absolutely. This is incorrect and leads us into many traps, such as trying to 'teach for understanding' as an absolute when understanding can be viewed as a continuum and the nature and degree of understanding sought should be part of a teacher's articulated curricular intent.

Von Neumann Architecture	A description of a computer architecture first described by John Von Neumann. It describes an electronic computer system comprising of input, processing, memory, secondary storage and output.
Working (short-term) memory	Where conscious processing or ‘thoughts’ occur. Limited to holding four to seven items of information for up to around 30 seconds at a time.