



love the journey

Curriculum Implementation 2025-26

Secondary

LCA Strand	Technology, Enterprise & Sport
Subject	Computer Science
Key Stage	Key Stage 4 (Chapter 10-11)

<p>What are the key concepts taught?</p>	<p>At Key Stage 4 in Computer Science, pupils will learn the following concepts:</p> <ul style="list-style-type: none">• Algorithms: This involves the creation of step-by-step instructions that a computer can follow to solve a particular problem. Students will learn how to create, test, and refine algorithms, as well as how to evaluate their efficiency.• Programming: Students will learn how to write code in a high-level programming language called Python. They will learn how to use variables, data types, conditional statements, loops, and functions to create programs that can solve problems.• Data: Students will learn how data is represented and stored in computers, and how to manipulate and analyse data using programming languages. They will also learn about the security and ethical considerations around the use of data.• Computer systems: Students will learn about the hardware and software components of computer systems, as well as how data is transmitted and processed across networks. They will also learn about the different types of computer systems and their applications.• Computer networks: Students will learn how computer networks are set up and configured, and how data is transmitted across different types of networks. They will also learn about the security and ethical considerations around the use of computer networks.• Cybersecurity: Students will learn about the threats and risks associated with using computer systems and networks, and how to protect against them. They will also learn about the legal and ethical issues around cybersecurity.• Computational thinking: Students will develop their problem-solving skills and learn how to break down complex problems into smaller, more manageable parts.
--	---

	<p>They will also learn how to design and test solutions to problems.</p>
<p>What is the sequencing of units?</p>	<ul style="list-style-type: none"> • Computational thinking, algorithms, and programming: This unit covers the fundamentals of computational thinking, including problem-solving, decomposition, pattern recognition, abstraction, and algorithms. Students will also learn how to write programs in a high-level programming language such as Python, as well as how to test and debug their code. • Computer systems: This unit covers the hardware and software components of computer systems, including the CPU, memory, storage devices, input and output devices, and operating systems. Students will also learn about the different types of software and their uses, as well as how to use the command line interface.
<p>How do we encourage pupils to see the links between different units and concepts?</p>	<p>Showing links between the different concepts taught helps to develop the pupils' critical thinking and problem-solving skills. Some methods used are:</p> <ul style="list-style-type: none"> • Real-world examples to show pupils the connection and to see how they are applied in the real world. • Visual aids: diagrams/mind maps/flowcharts • Class discussions to allow pupils to hear different opinions from their peers
<p>What are the planned opportunities for adaptive teaching, including for SEND, the more able, and disadvantaged pupils?</p>	<ul style="list-style-type: none"> • Real-time feedback – providing feedback to pupils as they're working through the lesson activities. • Varying the lesson pace, allowing pupils to work at their own speed through scaffolded resources or video tutorials. • Variety of teaching strategies to support different learners, i.e., hands-on activities to help pupils learn best through practical tasks. • Gamification of lesson activities to engage learning, particularly with disadvantaged or SEND pupils
<p>What are the planned opportunities for retrieval and reflection by pupils?</p>	<ul style="list-style-type: none"> • Do now / retrieval tasks at the start of the lesson to check previous understanding • Use of revision software like Smart revise gives pupils opportunities to test their knowledge of long answer questions, quiz questions, and key terminology. • Review quizzes: Like Seneca, Kahoot, or Quizlet can be used to recall information from previous lessons. • Peer feedback: pupils give their peers valuable feedback on tasks completed

	<ul style="list-style-type: none"> Exit tickets at the end of the lesson so that pupils can write a brief response to a question related to the concepts learnt in the lesson.
--	---

<p>What are the opportunities for feed forward by the teacher post assessment outcomes?</p>	<ul style="list-style-type: none"> In coding lessons, pupils are given specific recommendations for improving the syntax of the code, structure, or logic. In multimedia projects, pupils are given ways to improve the design or layout of the project Teachers regularly review the levels of attainment of each pupil to set them targets for the next progress phase. Pupils are also encouraged to set their own targets on the LC assessment record forms. Departmental data allows teaching staff to analyse and identify pupils' levels, which helps to inform future planning.
---	---

<p>What are the planned opportunities for developing Reading?</p>	<ul style="list-style-type: none"> Giving pupils the opportunity to read out loud, i.e., reading the instructions for an activity or reading a context statement. Key technical terminology is displayed for all pupils to see Using case studies to provide pupils with real-world examples of how technology is used in industry. Using online digital resources (e.g., online tutorials to provide pupils with interactive reading opportunities)
---	--

<p>What are the planned opportunities for developing literacy, numeracy, oracy and SMSC?</p>	<p>Literacy:</p> <ul style="list-style-type: none"> Pupils are allowed to write in a range of styles, like technical reports, instructions, and code documentation, using appropriate grammar and spelling. <p>Numeracy:</p> <ul style="list-style-type: none"> Pupils will learn numeracy through algorithm design, as it will develop their logical reasoning skills to solve problems. Using binary and hexadecimal allows pupils to learn new number systems and be able to convert between them. Teaching pupils to program helps to develop their ability to use mathematical concepts like variables and functions to solve problems. Students can learn about numerical concepts related to cybersecurity, such as encryption and decryption algorithms, checksums, and hashing algorithms. Encouraging pupils to solve real-world problems using mathematical skills. <p>Oracy:</p> <ul style="list-style-type: none"> By modelling clear and effective speaking in the teacher's own communication. Pupils have the opportunity to deliver presentations to the rest of the class Provision of regular feedback on pupils' oracy skills
--	---

- Pupils also get the opportunity to have paired/group discussions to help develop their speaking and listening skills.

SMSC:

- Pupils at KS4 get the opportunity to work on joint tasks/projects to help develop their social skills and to help them work effectively in teams. The main area where they get to work together is through paired programming challenges, which occur each week.
- Looking at a variety of computer science pioneers who all come from very different backgrounds.
- Pupils will explore algorithm bias in artificial intelligence and machine learning systems and discuss the implications of this.
- Pupils also look at the responsible and safe use of digital technologies, such as cyberbullying and digital citizenship.
- Pupils also look into open source software and intellectual property, exploring issues like software patents, copyright infringements etc