



love the journey

**Curriculum Implementation 2025-26**

**Secondary**

<b>LCA Strand</b>	<b>Technology, Enterprise &amp; Sport</b>
<b>Subject</b>	<b>BTEC Computing AAQ</b>
<b>Key Stage</b>	<b>Key Stage 5 (Chapter 12-13)</b>

<p>What are the key concepts taught?</p>	<ul style="list-style-type: none"> <li>• <b>Unit 1: Programming Fundamentals (Externally Assessed)</b> Students learn the building blocks of programming, including variables, data types, control structures, procedures/ functions, and error handling. Emphasis on algorithmic thinking and problem-solving using a high-level language.</li> <li>• <b>Unit 2: Computer Network Security and Encryption (Externally Assessed)</b> Explores types of cyber threats, vulnerabilities in networked systems, and key principles of cybersecurity. Students study encryption methods, authentication, firewalls, and develop strategies for protecting IT systems.</li> <li>• <b>Unit 3: Human-Computer Interaction (Internally Assessed)</b> Covers principles of user interface design, accessibility, usability, and user experience. Students evaluate existing systems and design their own user interfaces based on user needs and feedback.</li> <li>• <b>Unit 4: Practical Programming (Internally Assessed)</b> Students plan, develop, test, and evaluate a fully working software solution to meet a specific need. Emphasis is on clean code, version control, user feedback, and iterative development.</li> </ul>
--	---

<p>What is the sequencing of units?</p>	<p><b>Year One</b></p> <ul style="list-style-type: none"> <li>• Unit 1: Programming Fundamentals (6 lessons per fortnight)</li> <li>• Unit 3: Human-Computer Interaction (3 lessons per fortnight)</li> </ul> <p><b>Year Two</b></p> <ul style="list-style-type: none"> <li>• Unit 2: Computer Network Security and Encryption (6 lessons per fortnight)</li> </ul>
---	---

- Unit 4: Practical Programming (3 lessons per fortnight)

How do we encourage pupils to see the links between different units and concepts?

Programming principles in Unit 1 feed directly into the software project in Unit 4

- Knowledge of secure coding and encryption from Unit 2 can be applied in software design
- User-centred design from Unit 3 improves the usability of final projects in Unit 4
- Real-world scenarios help link abstract knowledge to everyday technology use

**Methods include:**

- Mind maps and concept linking activities
- Cross-referenced vocabulary and learning outcomes
- Scenario-based tasks that incorporate content from multiple units
- Reflection logs where pupils explicitly identify cross-unit links

What are the planned opportunities for adaptive teaching, including for SEND, the more and able and disadvantaged pupils?

- 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.
- Tiered programming challenges to support all abilities
- Sentence starters and coding skeletons for scaffolded learning
- Interactive tutorials and flipped learning videos
- Practical demonstrations followed by peer instruction
- Differentiated feedback and flexible success criteria in coursework units

What are the planned opportunities for retrieval and reflection by pupils?

- Do now / retrieval tasks at the start of the lesson to check previous understanding
- Do Now coding problems or security concept quizzes
- Weekly low-stakes tests and quick-fire recap questions
- Use of revision software like know-it-all all ninja gives pupils opportunities to test their knowledge of long answer questions, quiz questions, and key terminology.
- Past exam questions to prepare for the exam, but to also improve their recall
- Walkthrough videos and screencasts for self-review
- Reflective logs in programming tasks documenting what was learned and how they improved
- Provide personalised feedback for each student on their class notebooks. This is done at each phase of their coursework submission. Audio feedback is given to allow pupils to improve their work.

	<ul style="list-style-type: none"> <li>• Review quizzes: Like knowitallninja, Kahoot or Quizlet can be used to recall information from previous lessons.</li> <li>• Peer feedback: pupils give their peers valuable feedback on tasks completed</li> <li>• Encourage self-reflection: Encourage students to reflect on their learning and identify areas where they need further support.</li> </ul>
--	--

<p>What are the opportunities for feed forward by the teacher post assessment outcomes?</p>	<ul style="list-style-type: none"> <li>• Breakdown of coding errors and refactoring advice in Unit 4</li> <li>• Review sessions based on mock exam analysis in Units 1 &amp; 2</li> <li>• Marking rubrics shared with pupils in advance to clarify expectations</li> <li>• Post feedback action plans and self-assessment checklists</li> <li>• Data-informed intervention sessions and progress tracking</li> <li>• Teacher to set and review individual targets</li> </ul>
---	--

<p>What are the planned opportunities for developing Reading?</p>	<ul style="list-style-type: none"> <li>• Giving students the opportunity to read out loud, i.e. reading the instructions for an activity or reading a context statement.</li> <li>• Key technical terminology displayed for all pupils to see</li> <li>• Using case studies to provide pupils with real world examples of how technology is used in industry.</li> <li>• Using online digital resources (e.g. online tutorials to provide pupils with interactive reading opportunities)</li> <li>• Reading and interpreting code snippets and debugging logs</li> <li>• Reviewing and analysing case studies in HCI and cybersecurity</li> <li>• Keyword displays and vocabulary revision banks</li> <li>• Access to technical reading materials and online documentation (e.g. W3C, MDN)</li> <li>• Pupils reading aloud task descriptions or peer code reviews to develop technical fluency</li> </ul>
---	---

<p>What are the planned opportunities for developing literacy, numeracy, oracy and SMSC?</p>	<p><b>Literacy:</b></p> <ul style="list-style-type: none"> <li>• Students are given the opportunity to write in a range of styles, like technical reports, instructions, and code documentation, using appropriate grammar and spelling. Their coursework requires them to demonstrate their understanding of computing concepts and communicate their ideas effectively.</li> <li>• Peer reviews: During their coursework tasks students peer review each other's work, providing constructive feedback on content and style.</li> </ul> <p><b>Numeracy:</b></p> <ul style="list-style-type: none"> <li>• Students will learn numeracy through algorithm design as it will develop their logical reasoning skills to solve problems.</li> </ul>
--	--

- 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 students to solve real-world problems using mathematical skills.
- Logical reasoning, algorithms, and flowcharts
- Programming involving loops, mathematical operations, and coordinate systems

**Oracy:**

- By modelling clear and effective speaking in the teacher's own communication.
- Students have the opportunity to deliver presentations to the rest of the class
- Provision of regular feedback on students' oracy skills
- Students also get the opportunity to have paired/group discussions/programming to help develop their speaking and listening skills.

**SMSC:**

- Students 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 they get to work together is through paired programming challenges, which occur each week.
- Students 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.
- Students also look into open source software and intellectual property, exploring issues like software patents, copyright infringements etc