Computer Science - Learning Journey Year 2025-2026

Autumn Term

	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	
Topic	**1. 11 – Professsional Standards 1.1 Hardware 1.8 Algorithms and Constructs**			**2. Implementation 1.5. Data Organisation 1.8 Algorithms and Constructs 2. Investigation, Implementation & Testing 1.10 Program Construction**				**1. 11 – Professsional Standards 1.1 Hardware 1.5. Data Organisation 1.8 Algorithms and Constructs 2. Investigation & Implementation**	HALF TERM
Challenge Objective and Content (for all learners)	• Formal and informal codes of ethical behavior and conduct in computer science • Hardware and Architecture, factors effecting performance & embedded Systems			• Designing, writing, testing and refining Python 3 code including: Variables, Sequence, Selection, Iteration, Arrays, Lists and File Handling. • Experience and understand examples of programming errors.				• Recap and retrieval practice, revision and an EOT tEst.	
Inspire Opportunities	Physical hardware components will be investigated; students will get a hands-on approach to their learning through the medium of assembling hardware components and seeing what happens “under the hood”. To stretch and challenge students will complete More complex coding constructs, investigation and implementation tasks. Career related real world context research/questions to inspire students to broaden their understanding of computer science within their own environment.								
Assessment Opportunities	Formative assessments throughout utilising MCQ and google forms. End of topic summative assessments.								

	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	
Topic	**1.3 Networking and Cybersecurity 1.5. Data Organisation**				**1.8 Algorithms and Constructs 2. Investigation & Implementation**		CHRISTMAS
Challenge Objective and Content (for all learners)	• Network characteristics, topologies, hardware, protocols & metrics.	• The purpose of the Internet, DNS, URL’s hosting and the cloud.	• Factors effecting performance of networks.	• Cybersecurity threats and network protection.	• Python coding using the prior unit and incorporating this unit into coding challenges.	• Recaps and revision and an EOT test.	
Inspire Opportunities	Career related real world context research/questions to inspire students to broaden their understanding of computer science within their own environment. To stretch and challenge students will complete complex algorithms using python coding creating quizzes, file constructs and complete various coding challenges.						

Computer Science - Learning Journey Year 2025-2026

Assessment Opportunities	Formative assessments throughout utilising MCQ and google forms. End of topic summative assessments.	

Spring Term

	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	
Topic	**1.4 Data Representation and Storage** **1.8 Algorithms and Constructs** **2. Investigation & Implementation**						HALF TERM
Challenge Objective and Content (for all learners)	• Representation of Numbers: Binary, Binary Addition and Subtraction, Underflow and Overflow, Sign, Magnitude and Twos Complement, Hexadecimal.			• Storage Requirements: Units of measurement and conversions.	• Representation of Characters, Graphics and Sound	• Compression types and ratios	
Inspire Opportunities	Students will research what the number bases are specifically used for and conduct their own checks into conversions for colours, emoji's and logical representations.			Students will create code that will complete the conversations and calculations for them once they have learnt about the calculations themselves. This will inspire students to create coded solutions to theoretical aspects of computer science			
Assessment Opportunities	Formative assessments throughout utilising MCQ and google forms. End of topic summative assessments.						

	Week 1	Week 2	Week 3	Week 4	Week 5	
Topic	**1.5. Data Organisation** **1.8 Algorithms and Constructs** **2. Design, Investigation, Implementation & Testing**					EASTER
Challenge Objective and Content (for all learners)	• Functions, Procedures and Parameters • Designing effective testing strategies to ensure the final solution meets requirements. • Implementing effective testing using python 3.0		• Graphical user interfaces: Introducing Tkinter • Creating new and extend existing functions or methods. • Creating new and editing existing objects, buttons, labels, text boxes and message boxes. Commands Get, Entry, Destroy, Configure & Set.			

Computer Science - Learning Journey Year 2025-2026

	• Test data types • Handling user inputs • Validation and Verification	Students will begin by interrogating code that has already been constructed so they are able to see the trajectory of what they will be aspiring to complete by the end of the course. They will then continue forward to design and implement their own python programming GUI's using Tkinter.	
Inspire Opportunities	Implementation of theory by creating coded solutions with increasing difficulty to challenge all.		
Assessment Opportunities	Formative assessments throughout utilising MCQ and google forms. End of topic summative assessments.		

Computer Science - Learning Journey Year 2025-2026

Summer Term

	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	
Topic	**1.8 Algorithms and Constructs 2. Design, Investigation & Implementation & Testing 1.10 Program Construction**				**Coding Challenges and Mock Exam Week**	**1.8 Algorithms and Constructs Component 2 - Consolidation of Learning outcomes**	HALF TERM
Challenge Objective and Content (for all learners)	• Computational thinking including: Decomposition, Abstraction, Pattern recognition and computation thinking. • How to create and use flowcharts. • Planning a program using flowcharts. • Introduction to Pseudocode: How to write and use pseudocode. • Planning a program using pseudocode	• Programming from a plan to meet a set of requirements • designing and documenting input and output facilities for effective user interfaces • Be able to experience and understand examples of programming errors. Implementation of theory by creating coded solutions that increase with difficulty to challenge all.	Students will receive an exam scenario to work upon in class. This is to help familiarise the students with the content of component 2. Walking Talking Mock examination.		• Students will be continuing to consolidate learning that has taken place this year and apply this to a multitude of coding challenges depending on learning needs identified during the previous week. Recapping on concepts on component 1: theory-based aspects of the computer science GCSE. Revision time will be allocated, both directed and some independent time with mock examination occurring at the end of the week	Prep for Mock Exams Continuing with Tkinter practice. Combining all elements of python that have been taught thus far and introducing students to their exam format – on screen examination.	
Inspire Opportunities	Computational thinking can be utilised across multiple subjects, and this will be discussed at length during this section of the curriculum, cross curricula with DT and design that meets a set of requirements as well as the introduction of pseudocode which will be discussed as a way to plan a coding project before implementation.						
Assessment Opportunities	Formative assessments throughout utilising MCQ and google forms. End of topic summative assessments.						

	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	
Topic	**PLC's and Refinement**		**1.11 Impacts of Digital Technology on Wider Society**		**1.7 Principles of Programming 1.9 Software Development 1.10 Program Construction**		SUMMER

Computer Science - Learning Journey Year 2025-2026

Challenge Objective and Content (for all learners)	• Discussions from Component 1 mock examination, students will be receiving feedback with time for them to respond. • PLC Learning checklists will be updated and based upon the Component 1 mock examination. • This period will be used to recap and retrieve areas of the Component 1 theory that students have found difficult to understand as highlighted by PLC's and Mock examination.	• The ethical, legal, cultural, environmental and privacy issues linked to the use of computer systems. • The impact of relevant current legislation on Computer Science,	**Work experience week**	• the characteristics and purpose of high-level and low-level languages • situations that require the use of a high-level or a low-level languages. • Students to learn the technical terms and uses for Integrated Development Environment (IDE) tools.	• Investigate the IDE tools in practice whilst completing coding challenges. • Students to understand the purpose program translators: compilers, interpreters & assemblers •Learn and understand the five principal stages involved in the compilation process • Be able to experience and understand examples of programming errors.	
Inspire Opportunities	Implementation of theory by creating coded solutions with increasing difficulty to challenge all. Theoretical components combined through the use of revision materials.			Career related real world context research/questions to inspire students to broaden their understanding of computer science within their own environment.		
Assessment Opportunities	Formative assessments throughout utilising MCQ and google forms. End of topic summative assessments.			Formative assessments throughout utilising MCQ and google forms. End of topic summative assessments.		