Thomas Mills
High School & Sixth Form

CURRICULUM OVERVIEW: COMPUTER SCIENCE

Our Computer Science curriculum at Thomas Mills High School & Sixth Form aims to provide pupils with a broad understanding and depth of knowledge in computing, building directly upon the foundational skills and concepts introduced in Key Stage 3. We are committed to fostering computational thinking, problem-solving abilities and a deep understanding of how technology works, ensuring that pupils are not only informed users of technology but also creative and responsible digital citizens.

Through our curriculum, pupils will develop a strong understanding of various computing principles, from the representation of algorithms to the intricacies of computer networks and cybersecurity. Building on the algorithmic thinking and basic programming constructs learned in Year 9 (such as sequence, selection and iteration with pseudocode), Year 10 delves deeper into practical Python programming with units on advanced iteration (for and while loops). Furthermore, the foundational understanding of computer networks and cybersecurity from Year 9 will be expanded upon with detailed units on data representation, databases and the wider impacts of digital technology on society. We encourage analysis, evaluation and discussion throughout the schemes, providing opportunities for pupils to reflect on their own views and the impact of technology on society. There is a strong emphasis on respect, diversity of view and feeling empowered to share ideas and ask questions.

Our goal is to prepare pupils for their next steps, whether that be further academic study in Computer Science or related fields, apprenticeships, or future careers in the ever-evolving digital landscape. We equip them with the essential knowledge and practical skills to navigate and contribute positively to the digital world, empowering them to be confident, capable and ethical participants in a technology-driven society.

Qualification: AQA GCSE Computer Science 8525

Thomas Mills
High School & Sixth Form

Year 10

Term	Topic	Knowledge and Skills	Useful Links
1	**Unit 7:** Databases and SQL	This unit introduces the concept of databases, which are organised collections of information. They will learn how data is stored and how to retrieve, insert, update and delete information using a programming language called SQL (Structured Query Language). **Knowledge:** • What a database is. • Database key terms (table, record, field, primary key, foreign key). • A flat file database. • A relational database. • The function of SQL. • The function of different data types within databases. **Skills:** • Using SQL to retrieve data from a table in a relational database. • Using SQL to retrieve data from more than one table in a relational database. • Using SQL to insert, update and delete data into a relational database. • Interrogating and updating an existing database using knowledge of databases and SQL.	**How you can help at home:** • Discuss examples of databases in everyday life (e.g.- a library catalogue, online shopping websites, your contact list on your phone). • Talk about how companies manage large amounts of customer data. • If you use any simple personal databases (like a spreadsheet for recipes), discuss how information is organised. **AQA Specification:** • 3.7.1 to 3.7.2 • AQA | Computer Science | GCSE | GCSE Computer Science **Links:** • Ada Computer Science: Databases • Ada Computer Science: SQL • Craig n Dave: SLR14 - Relational databases and SQL • SQL Zoo • SQL Island Game • W3Schools: SQL Tutorials • PG Online Textbook: Chapter 7: 7.1 to 7.3 • Seneca Learning: Computer Science

Thomas Mills
High School & Sixth Form

Assessments	Your child will have a mid-point multiple-choice quiz and a summative assessment (made up of past paper examination questions) to evaluate their understanding of databases and SQL. Past paper questions will be a mix of 'shade the lozenge', short answer and longer answer questions.

Thomas Mills
High School & Sixth Form

CURRICULUM OVERVIEW: COMPUTER SCIENCE

1	**Unit 2E:** Learn how to use counter-controlled iterations (For loops)	This unit focuses on for loops, a way for computers to repeat a set of instructions a specific number of times. This is essential for automating repetitive tasks in programming. **Knowledge:** • What a for loop is and its purpose in programming to repeat actions. • How 'for loops' are structured and written in the Python programming language. • How to plan the steps (algorithm) for a program that uses repetition. **Skills:** • The ability to read and understand Python code that uses for loops. • Changing existing programs to alter their repetitive behaviour. • Writing algorithms in Python involving counter-controlled iterations. • Applying computational thinking to break down problems and use loops for efficient solutions.	**How you can help at home:** • Encourage your child to try out simple iterative Python programs using online interpreters such as Edublocks. • Discuss real-world examples of repeating tasks, like counting items. • Play simple online coding games that involve repeating actions. • Encourage them to think about how for loops could be used in everyday scenarios, such as generating a list of numbers or repeating a greeting. **AQA Specification:** • 3.2.2 • AQA | Computer Science | GCSE | GCSE Computer Science **Links:** • BBC Bitesize: Programming concepts - AQA • Ada Computer Science: Programming concepts • W3Schools Python Tutorial (A good beginner-friendly Python resource) • Craig n Dave: SLR08 - Basic programming concepts • PG Online Textbook: Chapter 2: 2A.3 • Seneca Learning: Computer Science

Assessments	Your child will have a mid-point multiple-choice quiz to evaluate their understanding of counter-controlled iteration.

1	**Unit 2F:** Learn how to use condition-controlled iterations (While loops)	This unit introduces while loops, another way for computers to repeat instructions, but this time based on whether a certain condition is true. **Knowledge:** • Understanding that iteration means repeatedly executing a group of instructions. • What a while loop is and its purpose in programming to repeat actions if a condition is true. • Knowing the differences between for loops and while loops and when to use each. • How while loops are structured and written in Python. **Skills:** • Defining iteration as a group of instructions that are repeatedly executed. • Modifying a program to incorporate a while loop. • Comparing a while loop and a for loop. • Writing algorithms in Python involving condition-controlled iterations. • Applying computational thinking to design programs that handle situations requiring repetition until a specific outcome.	**How you can help at home:** • Encourage your child to try out simple iterative Python programs using online interpreters such as Edublocks. • Discuss scenarios where something repeats until a condition is met (e.g. - keep stirring until the mixture is smooth). • Play games that involve repeating an action. • Encourage them to consider how while loops might be used in apps, like a game that keeps asking for input until the correct answer is given. **AQA Specification:** • 3.2.2 • AQA | Computer Science | GCSE | GCSE Computer Science **Links:** • BBC Bitesize: Programming concepts - AQA • Ada Computer Science: Programming concepts • W3Schools Python Tutorial (A good beginner-friendly Python resource) • Craig n Dave: SLR08 - Basic programming concepts • PG Online Textbook: Chapter 2: 2A.3 • Seneca Learning: Computer Science

Thomas Mills
High School & Sixth Form

Assessment	Your child will have a mid-point multiple-choice quiz and a summative assessment (made up of past paper examination questions) to evaluate their understanding of counter-controlled iteration (Unit 2E) and condition-controlled iteration (Unit 2F). Past paper questions will be a mix of 'shade the lozenge', short answer and longer answer coding-based questions.

2	**Unit 3:** Data Representation	This unit explores how computers store and represents all types of information, from numbers and text to images and sound. They will delve into the binary system (using only 0s and 1s) that forms the foundation of all digital data. **Knowledge:** • Why computers use binary (0s and 1s) for all data and instructions. • Understanding decimal (base-10), binary (base-2) and hexadecimal representations, including counting and conversion concepts. • How text (ASCII, Unicode), images (pixels, colour depth, resolution) and sound (analogue to digital conversion, sample rate, duration, resolution) are stored digitally. • The purpose of compression (lossy vs. lossless) and knowledge of methods like Run Length Encoding (RLE) and Huffman coding. • Understanding overflow and underflow errors in data representation. **Skills:** • Converting between binary, decimal and hexadecimal. • Performing binary addition and understanding binary shifts. • Calculating storage requirements for text, bitmap images and sound files based on their properties.	**How you can help at home:** • Discuss how digital photos or music files take up space on devices. • Explore online binary converters or games that involve binary numbers. • Talk about how streaming services use compression to deliver content quickly. **AQA Specification:** • 3.3.1 to 3.3.8 • AQA | Computer Science | GCSE | GCSE Computer Science **Links:** • BBC Bitesize: Fundamentals of data representation - AQA • Ada Computer Science: Representation of numbers • Ada Computer Science: Representation of text • Ada Computer Science: Representation of images • Ada Computer Science: Representation of sound • Ada Computer Science: Compression • Cisco Binary Game • Flippy Bit Game • Craig n Dave: SLR13 - Data representation • PG Online Textbook: Chapter 3: 3.1 to 3.6 • Seneca Learning: Computer Science

		• Applying and interpreting RLE and Huffman coding for data compression, including calculating compression ratios and building/interpreting Huffman trees. • Comparing storage requirements of compressed vs. uncompressed data.	
Assessments		Your child will have two mid-point multiple-choice quizzes and a summative assessment (made up of past paper examination questions) to evaluate their understanding of data representation. Past paper questions will be a mix of 'shade the lozenge', short answer and longer answer written questions.	

3	**Unit 2G:** Learn how to handle user input	This unit teaches your child how to create programs that can interact with users by taking input from the keyboard. They will also learn about different types of data and how to prevent common errors. **Knowledge:** • How to obtain information from a user via the keyboard in Python programs. • Understanding and differentiating between common data types: integer, real (float), Boolean, character and string. • The process of converting data from one type to another (e.g. - text to a number). • Defining runtime errors and the purpose of validation checks to prevent them. • How to design algorithms that incorporate and respond to user input. **Skills:** • Writing Python code to receive and process user input. • Correctly using and converting between different data types in programs. • Implementing basic validation checks to handle incorrect user input. • Creating algorithms and Python programs that are interactive and react to user provided data.	**How you can help at home:** • Encourage your child to try out simple interactive Python programs using online interpreters such as Edublocks (e.g. - a program that asks for their name and then greets them). • Discuss how everyday apps ask for and use your input (e.g. - logging in, searching). • Talk about the importance of entering correct information into forms online and what happens if you enter something wrong. **AQA Specification:** • 3.2.1 and 3.2.7 • AQA | Computer Science | GCSE | GCSE Computer Science **Links:** • BBC Bitesize: Programming languages - AQA • Ada Computer Science: Programming concepts • W3Schools Python Tutorial (A good beginner-friendly Python resource) • Craig n Dave: SLR08 - Basic programming concepts • PG Online Textbook: Chapter 2: 2A.1 • Seneca Learning: Computer Science

Assessment	Your child will have a mid-point multiple-choice quiz to evaluate their understanding of user input. Additionally, your child will have a summative assessment assessing all content taught to date, focusing on specification points 3.1, 3.2, 3.3, 3.5, 3.6 and 3.7. Past paper questions will be a mix of 'shade the lozenge', short answer and longer answer coding/written questions. This should be read in conjunction with the AQA subject content within the specification.

3	**Unit 8:** Impacts of Digital Technology	This unit explores the broader societal impacts of digital technology, including ethical considerations, legal frameworks, environmental effects and cultural changes. **Knowledge:** • Understanding the terms 'privacy', 'legal', 'ethical', 'environmental' and 'cultural' in a computing context. • Knowledge of data protection laws (e.g. - GDPR), the Freedom of Information Act and computer misuse offenses. • Awareness of the digital divide, broader social impacts of technology and ethical considerations including algorithmic bias. • Understanding the environmental effects of technology use. • The consequences of system downtime for organisations. **Skills:** • Applying ethical, legal, environmental and cultural concepts to computing scenarios. • Identifying copyright infringement and computer misuse offenses. • Recognising and explaining the social, cultural, ethical and environmental impacts of various technologies. • Participating in discussions and presenting on the impacts of technology.	**How you can help at home:** • Talk about recent news stories related to data breaches or ethical dilemmas in technology (e.g. - self-driving cars, social media's impact, etc). • Consider the environmental impact of old electronics and recycling. • Encourage critical thinking about information encountered online. **AQA Specification:** • 3.8 • AQA | Computer Science | GCSE | GCSE Computer Science **Links:** • BBC Bitesize: Ethical, legal and environmental impacts of digital technology - AQA • Ada Computer Science: Impacts and consequences • Ada Computer Science: Legislation • Craig n Dave: SLR06 - Ethical, legal and environmental impacts • PG Online Textbook: Chapter 8: 8.1 to 8.3 • Seneca Learning: Computer Science

		• Developing structured, well-reasoned responses for examination questions.	
Assessment		Your child will have a mid-point multiple-choice quiz, a presentation and a summative assessment (made up of a past paper examination question) to evaluate their understanding of the impacts of digital technology. The past paper question is a longer answer written question.	