# KS4 Computer Science Curriculum Overview

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | **Links to KS3** | In KS3, students learn about a different topic each team. These are specially selected to provide students with insights and experience into the different KS4 courses available. In KS3 Computer Science; students learn about computers, programming, data, algorithms, networks and cyber security. This early grounding provides students with the understanding to be able to first of all choose Computer Science at KS4 but also transition into this level of study. Networks, computational thinking, programming and data representation are main components of the Computer Science curriculum and are introduced in KS3. | | | | | |
| **Intent** | **Statement of Intent** | Students begin with Topic 3 as we believe that starting with the hardware of the computer is a logical and also interesting place to start with the students. Students will develop their knowledge in all 6 topics, each of which is different to each other but will be taught in a way that allows students to draw links between the topics. Students will receive fundamental and advanced programming skills that enable them to access questions for the exam but also prepare them for high level study of Computer Science or a career in programming. Students will revive a healthy balance of theoretical and practical application of computer science providing them with a breadth of knowledge and enthusiasm for the subject of Computer Science. | | | | | |
| | **Timeline** | **Term 1** | **Term 2** | **Term 3** | **Term 4** | **Term 5** | **Term 6** |
| **Implementation (Year 10)** | **Year Overview** | The objective in Year 10 is to give students the knowledge required for Paper 1 whilst also developing their programming skills for the on-screen Paper 2 Programming assessment they will sit in the Summer of Year 11. The Computer Science course is split into 6 units, in Year 10 students will develop in areas across all 6, but specifically: Computer, Data, Networks and Programming. This allows students to sit a mock in June for Paper 1. | | | | | |
| | **SOW** | **Systems Architecture and Programming** | | **Data representation and Programming** | | **Networks, Network threats and Programming** | |
| | **Assessment Type & Unit Focus** | 1 End of topic assessment comprised of a mixture of exam questions from this unit only - ***Teacher assessed*** Fortnightly Homeworks that consist of either past paper questions for this topic - ***Peer/self assessed*** Exam questions - **Peer/self assessed Or Seneca Learning - Automatically assessed**<br><br>Students learn about Architecture of the CPU, Memory, Cpu Performance, Secondary Storage. Weekly lessons will also comprise of Python Programming, developing through the different constructs of sequence, selection and iteration. | | 1 End of topic assessment comprised of a mixture of exam questions from this unit only - Teacher assessed Fortnightly Homeworks that consist of either past paper questions for this topic - Peer/self assessed Exam questions - **Peer/self assessed Or Seneca Learning - Automatically assessed**<br><br>Students learn about Units and Binary, Binary arithmetic and hexadecimal, Characters, Images, Sound, Data compression. Weekly lessons will also comprise of Python Programming, developing through the different constructs of sequence, selection and iteration. | | 1 mid and 1 end End of topic assessment comprised of a mixture of exam questions from this unit only - ***Teacher assessed*** Fortnightly Homeworks that consist of either past paper questions for this topic - ***Peer/self assessed*** Or Seneca Learning - ***Automatically assessed*** June Year 10 PPE week assessment will be comprised of a 1 hour vision of Paper 1. - ***Teacher assessed***<br><br>Students learn about the Internet and wide area networks, Local area networks, Wireless networking, Client-server and P2P networks, standards, protocols and layers. Students also learn about Network Threats and preventing vulnerabilities. In Weekly lessons will also comprise of Python Programming, developing an understanding of arrays, procedures, functions and files. |
| **Implementation (Year 11)** | **Year Overview** | Lessons will regularly start with retrieval activities to refresh and consolidate Year 10 knowledge before learning new content. As well as retrieval in lessons, students will be offered the option of a weekly session 6. In Year 11 we will focus on 3 topics in particular; Computational Thinking, Problem Solving with Programming and Issues and Impact. The first of those two topics will be taught in unison as the theory and practical side of computational thinking and programming allows learning to learn new information and apply and demonstrate this in an engaging way. | | | | | |
| | **SOW** | **Issues and impact** | Computational Thinking and programming | | **Logic and Languages** | | |
| | **Assessment Type & Unit Focus** | 1 end of topic assessment - Teacher assessed Fortnightly Homeworks that consist of past paper questions - Peer/self assessed d Paper 1 and Paper 2 mock - Teacher Assessed<br>Students learn about the topics of Ethical and cultural issues, environmental issues, legislation and privacy. | 2 mid term and 2 end of term topic assessments - Teacher assessed Fortnightly Homeworks that consist of past paper questions for Computational Thinking - Peer/self assessed h<br>Computational Thinking comprises of the concepts of decomposition and abstraction, algorithms and programming and truth tables. Students require this knowledge for Paper 1 as well as gaining programming and problem solving skills required for Paper 2. Students will also be developing their programming skills along side learning the theory of computational thinking. Students will build upon programming elements learned in Year 10 as well as learning new skills such as file handling and data structures. | | Paper 1 and Paper 2 mock - ***Teacher Assessed***<br>Students learn about Logic diagrams and truth tables, defensive design, errors and learning, translators and facilities, IDEs. | | |
| | **Enrichment Opportunities** | Coding club provides additional changes to learn coding and code breaking, | | | | | |
| **Impact** | **Assessment** | Assessment test after each unit. Assessments are followed by re-teach periods in which problem area are revisited. Tests are cumulative so content from all previous units appear in all tests | | | | | |
| | **Literacy and Numeracy links** | Y10: Students will develop their longer written answers for the topic of computers whilst developing their reading skills when reading about components in lesson.<br>Y11: Students will develop their numeracy skills and problem solving when programming. | | | Y10: The data unit will develop students numeracy skills through the ability of performing calculations and conversions in binary.<br>Y11: Further developing long written answers due to nature of this topic regularly being assessed this way. Students will research and read a lot around the topic. | | |
| | **How It Is Used / Skills Set Developed / Outcomes** | Students will learn a variety of different computer science elements that cover much of what is expected at high education. Students will develop their programming skills, primarily in Python where they will write advanced algorithms such as linear and binary searches as well as learning about low level languages through the use of the LMC simulator. Students will develop problem solving and computational thinking skills when trying to solve problems that require algorithms and/or programming. | | | | | |
| | **Careers in the Curriculum** | Throughout the course, we update our careers board using genuine examples of occupations. We also incorporate careers into starter activities looking at Computer Science related careers. | | | | | |