CS Department – Y7-Y13 Curriculum map

Year Group	Scheme of Work / Discipline	Knowledge	Skills &Techniques	Justification -
7	*Getting started*	• Logging onto the network, Teams, Email • The wider use of technology in everyday life • How to use PowerPoint to make a presentation • Know how to present	• Research skills to… • Identify one area that has been benefitted through the use of Computer Science • Presenting • Effective use of PowerPoint Assessment Students are assessed on their presenting skills	• Essential for all subjects – Login, emails • Presentation and PowerPoint skills are used across the school
	Excel	• Understanding why we use Excel • Understanding formulae (sum, if, average, max, min) • Understanding the use of conditional formatting	• Constructing a model in Excel • Appling formulae to a model • Appling conditional formatting to a model Assessment End of topic exam	• This is the introduction to coding in that the students gain experience of writing formula to control outcomes • It gives students the practice precision when writing code i.e. precise use of punctuation and syntax
	E-safety	• Understanding of e-safety • Understanding what appropriate / inappropriate behaviour online looks like • Knowing when and who to ask for help if something 'bad' happens • How to plan the construction of a video • How to edit video content	• Writing a script for a short film • Editing video • Acting • Filming	• E-Safety is essential at this age • Group work • A NC requirement
	Flowol	• Understand how flowcharts are used in CS • Flow control • Decisions • Loops • Subroutines • Logic (AND, OR)	• Use flowcharts to control devices • Incorporate loops, subroutines, and decisions correctly • Incorporate logic (AND, OR) correctly • Solve problem through decomposition	• This is the start of the students journey into thinking algorithmically • The flow of information in a flowchart needs to be understood before students can plan and construct a program
	Scratch	• Knowledge of the basic principles of programming • Variable declaration • If statement • Loop (forever, conditional, count controlled) • Sensing • Understanding the concepts of a game which make it challenging to play • Programming a maze game	• Giving variables descriptive names • Incorporating loops correctly • Incorporating if statements correctly • Applying sensing to the game • Designing and making a game	• Building upon the knowledge gained from Flowol, this is students introduction to coding • Blocks are used to remove the chance of syntax errors from their code, this cuts down on students feeling frustrated when their code does not work • Rapid progress can be made without the fear of making syntax errors • However, the chance of logical error remains
	Visual Studio, VB.NET and a	• Understand the basic concepts of text-based coding • Variables • Printing to screen • User input • If statement • Loops • Making a quiz • Scoring system	• Giving variables descriptive names • Incorporating loops correctly • Incorporating if statements correctly • Applying user inputs to the game • Designing and making a game	• Students use their knowledge of basic programming constructs and apply them to text-based code • The construction of the quiz is fairly straightforward however, students struggle to organise the code correctly to ensure the quiz works • Nevertheless, students have the skills learnt from Excel, Flowol, and Scratch to equip them to achieve well in this module of work

8	*Binary*	• Understand how computers represent numbers • Understand how a computer carries out simple mathematic operations (addition) • Understand how computers represent sound • Understand how computers represent characters • Understand how computers represent images/video	• Convert from denary to binary • Convert from binary to denary • Binary addition • Code/decode text to/from binary • Applying colour depth to create an image • Complete a digitisation of a sound wave	• Binary is a fundamental principle of CS and helps the student appreciate computational thinking from first principles
	Visual Studio, VB.NET and E-safety	• Understanding the basic principles of programming • How to create an app using console • Variable declaration • If statement • Loop (condition controlled) • To build a converter program with a user menu system	• Using Visual Studio • Coding with VB.NET • Basic programming concepts • Giving variables descriptive names using code • Incorporating loops correctly using code • Incorporating if statements correctly using code	• Building upon the text-based coding from Y7, this module asks the students to solve a more challenging task • The code can be extended to incorporate more challenging to convert units i.e. easy: miles to km, more difficult, Imperial height to metric, imperial weight to metric and vice versa
	Scratch	• Knowledge of the basic principles of programming • Use of random numbers • A menu • A scoring system	• Creating a maths-based game • Using the construction tools in scratch • Constructing a robust user menu	• This programming module builds upon the skills learnt in the previous Scratch module
	Photoshop – Airbrushing and digital tools	• Understanding that celebrities' photos have been altered using airbrushing • Understanding how to airbrush • How to use multiple digital assets to create a professional looking advertisement • Logo design	• Using photoshop • Downloading high quality and suitable images from the Internet • Designing a logo • Combining images and assets using Photoshop	• Improving the students' digital literacy • Teaching the students to be creating • Allowing the students to appreciate the work and skills involved in graphic design • Appreciate how and why the popular media manipulate images • Understand that manipulated 'beauty' is unobtainable and that people should not measure themselves to this fake standard
9	*Programming*	• Understanding the basic principles of programming • Variables, datatypes, loops, arrays, string manipulation, text files • Basic algorithms including: • Linear search – programmed • Binary search – teacher led program • Bubble sort – programmed • Insertion sort – programmed	• Programming with VB.NET through Visual Studio (IDE) • Problem solving • Abstraction • Decomposition • Computational thinking • Various problems solved by the students	• This year builds upon the programming knowledge from Y7 & Y8 •

10	Programming	• Understanding the principles of programming • Variables, datatypes, loops, arrays, string manipulation, text files • Basic algorithms	• Programming with VB.NET through Visual Studio (IDE) • Problem solving • Abstraction • Decomposition • Computational thinking	• All students must be given the opportunity to undertake a programming task(s), either to a specification or • to solve a problem (or problems), during their course of study. Students may draw on some of the content • in both components when engaged in Practical Programming.
	Paper 1	• Theory of computer systems: • Systems Architecture • Memory • Storage • Wired and wireless networks • Network topologies, protocols	• Examination techniques • AO1, AO2 • Unit tests • Dirt Understand and apply the fundamental principles and concepts of Computer Science, including abstraction, decomposition, logic, algorithms, and data representation understand the components that make up digital systems, and how they communicate with one another and with other systems. Apply mathematical skills relevant to Computer Science. Understand the impacts of digital technology to the individual and to wider society	• This part of the course follows the specification
	Paper 2	• Theory • Programming techniques • Producing robust programs • Computational logic • Translators and facilities of languages • Data representation	• Examination techniques • Solving department made programming challenges • analyse problems in computational terms • through practical experience of solving such • problems, including designing, writing and • debugging programs	•
11	Programming	• Theory of computer systems: • Systems Architecture • Memory • Storage • Wired and wireless networks • Network topologies, protocols	• Programming with VB.NET through Visual Studio (IDE) • Problem solving • Abstraction • Decomposition • Computational thinking	•

	Paper 1	• Theory of computer systems: • Systems Architecture • Memory • Storage • Wired and wireless networks • Network topologies, protocols	• Examination techniques • revision	•
	Paper 2	• Theory • Programming techniques • Producing robust programs • Computational logic • Translators and facilities of languages • Data representation	• Examination techniques • revision	•
12	Paper 1, 2 and coursework	Theory • Fundamentals of programming • CPU – Registers • CPU – Fetch, Decode, Execute • Types of processor • Input, output and storage • System software • Application Generation • Software development • Types of Programming Language • Compression, Encryption and Hashing • Databases • Networks • Web Technologies Programming Data types Data structures Boolean Algebra Computational Thinking Programming techniques Computational methods Algorithms Programming project Completing a teacher led project example using ASPX, Access and SQL	• Understanding the principles which form the foundations of CS • Programming fundamentals through to more advanced programming • Students will be expected to analyse, design, develop, test, evaluate and document a program written in a suitable programming language. The underlying approach to the project is to apply the principles of computational thinking to a practical coding problem. Students are expected to apply appropriate principles from an agile development approach to the project development.	Computer systems component (01) contains the majority of the content of the specification and is assessed in a written paper recalling knowledge and understanding. Algorithms and programming component (02) relates principally to problem solving skills needed by learners to apply the knowledge and understanding encountered in Component 01. Programming project component is a practical portfolio-based assessment with a task that is chosen by the teacher or learner and is produced in an appropriate programming language of the learner's or teacher's choice. Mathematical skills are embedded throughout the content of the three components. They will be assessed in the written papers and through the non-examined assessment where appropriate. The quality of extended responses are assessed in the written papers where indicated by an asterisk. It is marked using levels of response style mark schemes and in the Evaluation section of the Programming project component.
13	Term1	Coursework Preparation for mocks	Students will tackle a programming project of their choice	Programming project component is a practical portfolio-based assessment with a task that is chosen by the teacher or learner and is produced in an appropriate programming language of the learner's or teacher's choice.
	Term2	Mock preparation Exam preparation Completion of coursework		
	Term 3	Exam preparation	Past papers Revision guides Guided revision lessons	