UoB School Curriculum Outline – COMPUTER SCIENCE 2017/2018

Year	1A	1B	2A	2B	3A	3B
7	**Computer History.** Students will start learning about how fast the CPU has developed over the last 20 years which has enabled a massive growth in the complexity of software. **Introduction to algorithms.** Students will learn what an algorithm is and how to draw them using flowcharts.	**Introduction to programming.** Students will learn how to do simple programming using bitsbox which is coding website aimed at children. It covers all of the main aspects of coding including selection and variables. The focus of this unit is to familiarize students to text based coding and some of the underlying concepts they will need in year 8.	**Binary** A fun little unit where students learn to make any number they wish using just 1 and 0. **Hardware** Students will learn the purpose of a number of key aspects of computers. This includes the CPU, memory and storage devices.	**Robotics and computational thinking.** Problem solving is a core aspect of CS. One of the first skills students will need to learn is decomposition. In this unit students will be asked to solve problems using a robot. It will require them to decompose the problem and work in teams to find solutions.	**Bitsbox game** This unit students will be guided towards making their first game using bitsbox. They will use online video tutorials to get the bare bones of a game working. They will then add more features to the game using the skills they have developed so far.	**Revision and end of year assessment.** The end of year assessment will be split over two sessions. The first will explore theory whist the second will be a practical programming task. **Digital safety** This unit will be based on the current dangers young people face online. It will be focused on how they can show their virtues online.

8	**Problem solving.** This unit gives students the opportunity to develop their computational thinking skills by looking at a number of classic CS/mathematic problems. These include towers of Hanoi and 8 queens. **Sequencing and variables in python.** In year 7 students will have had their first taste of text based coding. This unit will help students learn the key concepts as well as learning a new programming language. Python is what students will learn all the way through school.	**Websites and website development.** Students are used to using websites, but not so used to creating them. This unit will explore some of the ideas behind websites as well as learning some HTML.	**Selection in python** Students have already been introduced to the idea of selection and in this unit they will explore how selection can be used to solve problems. This unit students will start to learn how to code without lots of help and guidance in preparation for the style of coding used at GCSE.	**Hacking and networking.** Students will have already explored some networking ideas when studying HTML. In this unit they will look at how computer networks are set up, the different types and how hackers try and bypass security measures.	**ASCII and recapping binary.** ASCII is how the computer stores letters using binary. Students will get an understanding that everything on a computer is stored using a clever sequence of 1 or 0.	**Revision and end of year assessment.** The end of year assessment will be split over two sessions. The first will explore theory whist the second will be a practical programming task. **Project – Making a simple computer game.** Students will make a game, showcasing the skills they have developed so far. This unit can be done in either scratch, game maker or python.

9	**Sequencing, selection and variables in python.** In year 8 students learned about small basic. This unit will take the knowledge gained in year 8 from small basic and translate it into python.	**ASCII and recapping binary.** ASCII is how the computer stores letters using binary. Students will get an understanding that everything on a computer is stored using a clever sequence of 1 or 0. **Introduction to machine learning** This is a short unit looking at how AI is changing the way we view computing. Students will explore different AI's and look at some of the basic concepts of machine learning.	**Iteration in python** In programming there are three main elements which make up almost every program. Variables, selection and iteration. In this unit students will learn how to make their code repeat which will allow them to develop more advanced programs and solve harder problems.	**Protecting your data** This unit will explore how data is encrypted online. Students will get a practical look at how messages are hidden, the different ways they can be hidden and ways to break simple encryption. **Images** In this unit students will learn how binary can be used to save something as complex as an image. They will understand how an image is made up of pixels, what image resolution means and how colour is represented by the computer.	**Legal and ethical aspects of CS.** Students will find themselves debating some of the big issues of the day. Who owns your data? Is it right for government to be able to read your email? Should end to end encryption be banned?	**Revision and end of year assessment.** The end of year assessment will be split over two sessions. The first will explore theory whilst the second will be a practical programming task. **Project** – Make your own python app

	Term 1a	Term 1b	Term 2a	Term 2b	Term 3a	Term 3b
12	**Programming –** Students will cover the following core aspects of programming. One of the key challenges for the first term is to get students who have never coded before to the right standard whist simultaneously developing the skills of the veteran coders. • Variables and expressions. • Selection • Iteration. • Arrays. **Theory topics -** • Details of the CPU. • RISC and CISC • Input, output and storage. • Binary	**Programming –** Students will finish their core learning with string manipulation and then start work on developing their computational thinking. Decomposition and thinking ahead are the two main themes of this term. **Theory topics -** • Boolean algebra. • Legal and moral impacts of CS. • Operating systems.	**Programming –** Students will further develop their computational thinking by exploring and using abstraction. They will also expand their coders toolset by exploring functions, local and global variables and using an IDE. **Theory topics -** • Networking. • Web technologies including HTML, CSS and JavaScript.	**Programming –** Students will further develop their programming by looking at standard algorithms and data structures. They will learn to program them from first principles and use them to solve problems. **Theory topics -** • Databases • Stacks • Queues • Bubble and insertion sort. • Binary and linear search.	**Programming –** In the final term before the exam students will round of their learning with file handling and low level programming in LMC. In this term more emphasis will be placed on exam style coding. **Theory topics -** • Software methodologies. • Revision.	On return after exam leave students will get started on the coursework for year 13. They will pick a project and make good headway into analysis. Students will also start to learn object orientated programming.

13	Students will continue their exploration of Object orientated programming. OO needs to be mastered before coursework coding can start. The main bulk of theory will be looking at complexity theory and algorithms. The new algorithms being introduced are A*, Dijkstra's shortest path, quick and merge sort. Students will also explore bitwise manipulation. **Coursework** Focus will be on design and starting the first prototype.	**Theory topics -** - Data structures. - Uses of hashing. - Computer methodologies. - Recursion. - Boolean algebra, adders and flip flops. **Coursework** Students will complete the coding and write up for the first prototype. The second prototype will be started and worked on over Christmas.	**Theory topics -** - Databases. - Pipelining. - GPU. - Thinking concurrently. - Caching. **Coursework** Students will complete prototype 2 and get started with prototype 3.	**Theory topics -** - Compilers. - Addressing modes. - Network security and encryption. - Server side technologies. - Page rank. **Coursework** Students will complete prototype 3 and then work on the Evaluation. The final deadline for the coursework will be immediately after Easter.	**Theory topics -** - Floating point addition. - Revision. **Coursework** Coursework will be submitted to the exam board.	NA