| Curriculum Map | | | | | | |
|---|---|---|---|---|---|---|
| Year 9 | | | | | | |
| Half term | Unit title with hyperlink to scheme of work | Unit summary | Skills & content covered | Skills & content revisited | Summary of formative marking, feedback and student response | Summative assessment schedule, including assessment criteria |
| Autumn 1 | Data Representation | Students will be equipped with the essential knowledge relating to data representation. Lessons will range from binary/denary/hex conversions to how sound and image are stored in a computer. | Convert denary to binary and vice versa. Carry out hexadecimal conversaions. Add/multiply/divide two binary integers together and explain overflow errors. Understand the use of binary representation using ASCII and Unicode. Explain how an image is represented as series of pixels in binary. How sound can be sampled and stored in digital form. Understand and explain different types of compression. | Students will be revisiting content learnt in Y8 and adapting their knowledge further in other types of data e.g. sound and image | Homework, Teams activities/tasks and verbal feedback. Identifying and correcting common misconceptions. | End of unit assessment |
| Autumn 2 | Logic gates, computational thinking and algorithm | Understand and apply the fundamental principles and concepts of logic. Students will learn and understand how to use AND, OR and NOT logic gates. Students will go on to develop an understanding of and apply the fundamental principles and concepts of abstraction, decomposition and algorithms. | This topic will focus on developing student understanding of Boolean logic by learning the theory of logic gates and their truth tables. Pupil's will also develop an understanding of simple logic diagrams using the operators AND, OR and NOT. Learners will also develop an understanding of combining Boolean operators using AND, OR and NOT logic gates and applying logical operators in truth tables to solve problems. Learners will develop and understanding of computational thinking such as abstraction, decomposition and algorithmic thinking and apply this theory to a range of problems. Students will apply the theory learnt to creating algorithms in forms of flow charts and pseudocode. | Pupil's will use their knowledge learnt in different programming languages, which was studied in Y8, to support a development of their skill set in different types of algorithms. They will apply this theory by creating a variety of flowcharts and pseudocode from given scenarios (some will be reflective of real life settings). | Homework, Teams activities/tasks and verbal feedback. Identifying and correcting common misconceptions. | End of unit assessment |
| Spring 1 | Python/algorithmic thinking | Students will spend critical time to analyse problems in computational terms through practical experience of solving such problems, including designing, writing and debugging programs using Python Programming. | This topic will introduce students to Python programming. Learners will learn essential programming fundamentals. Students will develop their understanding of programming fundamentals such as the use of the use of print(), input(), variables, constants, operators and assignments in coding. Pupil's will also develop a understanding of the use of constructs such as sequence, selection and iteration to control the flow of a program. | Learners will apply their skill set and knowledge learnt in Autumn 2, to start creating programs in Python. | Homework, Teams activities/tasks and verbal feedback. Identifying and correcting common misconceptions. | Students will be assessed by creating various programs using Python programming for given scenarios. |
| Spring 2 | Network | In this unit students will start to build an understanding of how computers can be networked together to perform tasks and to share data and resources. They will build an understanding of different network topologies then develop their understanding of how the internet works. | Students will be learning key terminology on how devices are connected together and how data is shared through a network. They will be able to explain the differences between LANs and WAN. Learners will be able to identify topologies and justify the appropriate one to use for a given scenario. Learners will be able to explain the role of different hardware needed to connect to a network and its role in transferring data. Students should be able to explain the different stages of packet switching and how users access websites (how the DNS system works). | During KS3 students completed 'Computer systems' unit, where students learnt the basics of connecting devices to a network and the purpose of some of the hardware reuiqred to build a network. Therefore, they have some understanding of the basic principles that underpin this area of the GCSE course. | Homework, Teams activities/tasks and verbal feedback. Identifying and correcting common misconceptions. | End of unit assessment |
| Summer 1 | System software & security | In this unit students will learn about the different categories of threats faced on a network and how individuals and organisations can directly address and prepare for these threats. Students will also go on to understand the key software required to run a computer. | Learners will be learn about social engineering techniques to gain access to user's personal data and a wide range of different threats to a network such as malware, brute force attack, denial of service etc. Pupils will then learn on how to | Pupils will be enhancing their knowledge on the previous unit of Networks. They will be using their understanding of how networks work in order to then protect it, usign physical and software based security. | Homework, Teams activities/tasks and verbal feedback. Identifying and correcting common misconceptions. | End of unit assessment |
| Summer 2 | Python | Students will go back to Python programming to secure the basic programming fundamentals before going into Y10. | Students should be able to create programs that include the following: print(), input(), variables, if statements and iteration. Students should understand the need for casting and how to use arithmetic operators. Pupils are to know boolean operators and comparison operators. They should use them accordingly in their code. | Learners will be focusing on enhancing and securing their programming skills and knowledge. Students will be tackling scenarios to create programs using previous knowledge and skill set of computational thinking. | Homework, Teams activities/tasks and verbal feedback. Identifying and correcting common misconceptions. | Students will be assessed by creating various programs using Python programming for given scenarios. |