Subject: Computing Year 9 Ability Mixed

	Unit 1	Unit 2	Unit 3
Topic	The Ethics of Computing	Algorithms	Python Programming - Sequence
Topic overview **Pupils will learn…**	Explore the legal, environmental and moral issues that now exist in our world as a result of the growing importance of technology in our daily lives and society	Understand the fundamentals of computational thinking (abstraction, decomposition, pattern recognition and algorithms) to solve a range of problems through flow diagrams and text-based programming	Understanding, through the use of text-based programming language, how to develop a range of simple programs, troubleshoot common issues and be able to identify appropriate data types and techniques to use within a given scenario
Components	• Understand the importance of crediting content creators and know how to find out who originally created material posted online in order to have a better understanding of how to use the internet responsibly. • Explore the difference between illegality and immorality and consider that technology is not immoral, but the way people use it can be in order to identify how to protect themselves from people who want to use technology immorally. • Know that e-waste has a negative impact on the environment (such as mining the minerals used in mobile phones fuels conflict, using digital devices uses a considerable amount of energy, contributing to climate change) in order be able to describe the benefits and drawbacks of one environmental impact and how the drawbacks might be minimised. • Explore the main laws that govern data protection in the UK in order to show understanding of GDPR and the Data Protection Act 2018 • Understand that driverless cars create moral dilemmas and design algorithms to program a driverless car to make moral decisions in order to be able to demonstrate computational thinking and consider the moral decisions that will be made by computers in the future.	• Understand the concepts of abstraction and decomposition in order to be able to focus on the most important aspects of a problem and break down large problems into manageable chunks • Design algorithms to solve a range of computational problems • Analyse the effectiveness of different approaches to solving problems in order to identify and apply the most efficient techniques when creating algorithms • Combine the principles of abstraction, decomposition and algorithm design with pattern recognition to solve a range of problems in order to replicate the user of algorithms to solve real world problems • Understand the benefits of a modular approach to programming to be able to develop efficient solutions for specific aspects of a larger problem • Recognise the standard symbols used in flow diagrams be able to read flow diagrams in order to visualise a solution to a problem and be able to design algorithms in the form of flow diagrams	• Understand a range of basic programming constructs in Python, including: know how to print to the screen, perform calculations, take inputs and store them in suitably named variables in order to navigate the programming language and develop basic programs that solve specific problems in the correct sequence • Develop working programs in Python to solve specific problems in order to show confidence in using its interface and understanding the correct use of syntax (rules) when writing program code • Analyse problems in computational terms, analyse the requirements of a program, identify the processes needed to solve a problem and plan creative solutions to problems in order to show understanding of computational thinking (from the previous unit) and apply it in a practical setting • Be able to read samples of Python code in order to spot errors and fix them, strengthening a fluency in use of the language in a range of unseen scenarios.
What pupils should already know (Prior learning components)	Throughout **Year 8 Computing**, students should have been taught to – • undertake creative projects that involve selecting, using, and combining multiple applications, preferably across a range of devices, to achieve challenging goals and meeting the needs of known users • Recognise basic concepts around laws that govern computing such as Copyright, Data Protection and Computer Misuse, and identify scenarios where these laws have been breached • Considered the positive and negative impacts of an increasing use of computers and devices on our daily lives from the perspective of a young person In **KS3 Geography**, students should have been taught to –	Throughout **Year 8 Computing**, students should have been taught to – • create, reuse, revise and repurpose digital artefacts for a given audience, with attention to trustworthiness, design and usability With **Year 7 Computing**, students should have been taught to – • understand basic key algorithms that reflect computational thinking and use logical reasoning to compare alternative algorithms for the same problem • used a block-based programming language to solve a variety of computational problems • examine a set of basic flow diagrams in order to determine the sequence of an algorithm and predict outcomes (As well as recognise the basic purpose of each shape)	Within the KS3 Computing, students should have been taught to – • create, reuse, revise and repurpose digital artefacts for a given audience, with attention to trustworthiness, design and usability • understand basic key algorithms that reflect computational thinking and use logical reasoning to compare alternative algorithms for the same problem • used a block-based programming language to solve a variety of computational problems and identify the roles played by specific aspects of code • examine snippets of code and translate what is happening in simple steps • understand the concept of sequencing within programs / algorithms and be able to place a set of instructions into the most appropriate order to solve a specific problem

	• understand how human and physical processes interact to influence, and change landscapes, environments and the climate		
Transferrable knowledge (skills)	• Being able to access computer systems, navigate to specific files and organise work in a logical structure. • Being able to use multiple pieces of software (such as a web browser, presentation software, word-processing software and a cloud computing system) in quick succession to create and refine design projects • Use of inference and articulation to obtain key knowledge from a topic and apply understanding when presenting findings • Being able to make reasoned arguments about an issue, having examined a set of views and facts, and categorised them as either positive and negative • Being able to consider the links between British Values and the conduct of individuals and institutions or organisations	• Being able to access computer systems, navigate to specific files and organise work in a logical structure. • Being able to use multiple pieces of software (such as a web browser, presentation software, word-processing software and a cloud computing system) in quick succession to create and refine design projects • Use of inference and articulation to obtain key knowledge from a topic and apply understanding when presenting findings • Being able to break down a large problem into smaller aspects and remove unnecessary details to identify key aspects	• Understand how to use sequence within a programming language to write a program in Python that follows a set of steps in a specific order to determine an outcome • Understand how to use a calculation within a program in Python that makes a mathematical calculation • Develop their understanding of the syntax within Python and correctly apply it to a range of scenarios • Explore the use of assignment, variables and data types to store data within a program and use it later on in a sequence within an algorithm • Be able to introduce random and time modules within Python to apply gaps of time within sequence and create programs that will generate random names or numbers
Key vocabulary pupil will know and learn	Credit, copyright, Creative Commons licence, illegal, immoral, digital footprint, e-waste, illegal, GDPR, Data Protection Act 2018, moral dilemma, algorithm, flow diagram	computational thinking, decomposition, abstraction, algorithm design, algorithm, pattern recognition, input, output, flow, process, decision	Sequencing, Variable, Assignment, Value, Input, Output, Syntax, Arithmetic Operators
Assessment activities	• Regular low stakes testing at the end of each lesson to check knowledge. • Practical lesson activities with digital activities assessed by teachers • Do Now tasks which test previous learning and build recall on key terms and applying them to specific contexts	• Regular low stakes testing at the end of each lesson to check knowledge. • Practical lesson activities with digital activities assessed by teachers • Do Now tasks which test previous learning and build recall on key terms and applying them to specific contexts	• Regular low stakes testing at the end of each lesson to check knowledge. • Practical lesson activities with digital activities assessed by teachers • Do Now tasks which test previous learning and build recall on key terms and applying them to specific contexts
Resources available	KS3 NC information National Curriculum - Computing key stages 3 and 4 (publishing.service.gov.uk) BBC Bitesize reference for laws and ethics in Computing Computers and the law - The law and ethics - KS3 Computer Science Revision - BBC Bitesize BBC Bitesize reference for Environmental Impact Energy use - Environmental impact - National 5 Computing Science Revision - BBC Bitesize BBC Bitesize reference for real-life issues around driverless cars PESTLE - Technological - Critical thinking and problem solving - WBQ National: Foundation KS4 Revision - BBC Bitesize	KS3 NC information National Curriculum - Computing key stages 3 and 4 (publishing.service.gov.uk) BBC Bitesize reference to computational thinking Computational thinking - KS3 Computer Science - BBC Bitesize BBC Bitesize reference to flowcharts Flowcharts - Designing an algorithm - KS3 Computer Science Revision - BBC Bitesize	KS3 NC information National Curriculum - Computing key stages 3 and 4 (publishing.service.gov.uk) BBC Bitesize reference to flowcharts Flowcharts - Designing an algorithm - KS3 Computer Science Revision - BBC Bitesize BBC Bitesize reference material and tutorials for Python programming Programming - KS3 Computer Science - BBC Bitesize
Notes **Why this topic is important…**	This unit will give students the opportunity to investigate some of the wider ethical issues surrounding modern information technologies, including the moral, environmental and legal issues that can arise in the digital age and will encourage them form opinions and develop arguments (ending with an in-depth case study looking at the moral dilemmas associated with driverless cars). This unit will introduce the concept of online content being owned and understand that material from the internet cannot be used without adequately crediting the creator. Students will use Google Search to track down the authors of online files. This expands on the concept of copyright infringement explored in **Year 8** by considering the method of crediting authors and ownership rather than simply paying for access. By examining key laws students will be able to assess how their own data must be protected and only used for	The term 'algorithm' has come to refer to any set of rules that precisely define a sequence of operations, such as making a cup of tea or cleaning your teeth. In the world of computing, an algorithm is a set of instructions that can be implemented as code to program a computer. Computational thinking is a logical, strategic approach to problem solving involving four cornerstones: decomposition, abstraction, pattern recognition and algorithm design to formulate an efficient and effective algorithm. These principles are key to being able to explore a problem and design an effective algorithm to solve it. Following on from an introduction to algorithms in **Year 7 unit 5**, this unit introduces students to the use of computational thinking to solve problems. They will learn about three of the four cornerstones of the computational-thinking approach to problem solving: decomposition,	There is a computer program behind just about everything we use today. Without computer programs, many things, from washing machines to aeroplanes, would not have the technological capabilities we have come to rely on. unit introduces students to writing a computer program in Python and covers taking inputs from the user, storing them in variables, calculating values using basic arithmetic operators and producing formatted output. The major data types are introduced, along with the key arithmetic operators needed to perform simple calculations in Python. The unit also looks at the concept of a list to store and manipulate multiple data items in Python and the basic manipulation of strings. The key programming construct underpinning all the work in this module is sequencing, which forms one of the three constructs that students will need to understand within GCSE Computer Science, and to be successful at programming more broadly.

	specific purposes, as embed the laws they must follow as citizens using the internet responsibly. It is important the students are encouraged to think about the difference between immorality and illegality and how technology enables people to do things that are not illegal but could be considered immoral. Students are encouraged to form their own opinions to develop their critical thinking with access to a number of case studies. The unit extends into environmental concerns, which links well with Geography and allows students to form views on how the use of technology and devices impacts the world. Students are also introduced, through a thought experiment, to the idea that programmers writing algorithms that control driverless cars are having to wrestle with very real moral dilemmas. Students are given two driverless-car scenarios to think about and through designing their own algorithms, will be able to consider the consequences actions programmed into systems can have. Writing algorithms at this point will support students when they explore them in more detail in **Unit 2 and 3 of Year 9**. Legal, ethical and environmental concerns form a significant portion of ideas explored within **GCSE Computer Science,** and will also support students in developing strong moral views in line with British values.	abstraction and algorithm design. They will then apply what they have learnt to create a number of different algorithms, which will support them to develop logical thinking skills and support them in understanding programming and algorithm design in **GCSE Computer Science**. The unit also builds upon the use flow diagrams, first introduced in the **Year 7 unit 5**, and students will learn how to use the functions of some standard flow diagram symbols and how to link them together to represent an algorithm. They will apply this learning to practical activities that will enable them to practise flow-diagram design and consolidate their comprehension of flow diagrams and how they relate to algorithms. It also links to **units 1 and 2 of Year 8** by identifying which components of the computer process data and the ways in which computers process data within **Year 8 unit 4**. Programming forms one of two major components within GCSE Computer Science and as such is a core element of KS3 Computing. This unit will enable students to program in a text-based language, and build upon smaller programming activities from using a block-based language within **Year 7 Computing.**	Throughout this module, students will use Repl.IT, a responsive online environment, to write and test their own code to solve coding challenges and develop their programming skills. There are opportunities to share code, working in groups to collaborate and trouble shoot example programs. These skills are fundamental to the programming industry within Computing, as well as supportive of the expectations that students can develop solutions in Python within **GCSE Computer Science**. Linking back to previous content, it builds upon **algorithms unit 2 in Year 9**, and ensures that students experience programming in two different languages (including **scratch in Year 7**.)