PONTEFRACT ACADEMIES TRUST

Computing: Years 7, 8 and 9; Computer Science: Years 10 and 11

		Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
Year 7	**Topic**	E-Safety	Micro:bit	Computational Thinking	Binary	PAT Travel - Part 1	PAT Travel – Part 2
	Enquiry Question	Is the internet a safe and inclusive environment?	How can we develop coding skills working with materials and electronics?	How can we learn to solve computational problems efficiently?	How do computers make decisions and perform instructions?	How can we use software to conduct effective research and planning?	How can we use software to produce high quality information?
	Key Knowledge and skills	• To describe ways in which we can stay safe online • To describe the most effective ways to create and store passwords • To share and present with others	• To describe iteration • To describe what a print statement does • To be able to run code on a micro:bit • To be able to create a 'for' loop • To be able to create a 'while' loop	• To describe what the four branches of computational thinking are • To be able to solve computational thinking problems	• How to convert denary, binary and hexadecimal • How to perform binary number addition • To calculate an output from logic circuits and to state an output from logic gates	• How to format text in word processing and spreadsheet applications • How to cut, copy, paste and edit work • How to use QWERTY keyboard commands • Digital literacy - how to formally use email	• How to format text presentation and web publishing applications • How to cut, copy, paste and edit work • Digital literacy - how to formally use email • How to create effective presentations • How to publish web content
	End Point	Students are aware of the dangers when browsing the web and how to use a mobile device safely Students can list and describe ways in which to stay safe online	Students can use coding skills to build their own programs using a micro:bit Students can analyse and annotate their code, finding errors and areas of improvement	Students can apply computational thinking to a range of problems when using a computer	Students understand how numbers are represented in binary and hex and can carry out simple operations on binary numbers Students understand simple Boolean logic [AND, OR and NOT] and some uses in circuits and programming	Students are confident and proficient in navigating and utilising Google applications (Docs, Sheets and Slides)	Students are confident and proficient navigating and utilising Google applications and can work within the four main strands of G Suite
Year 8	**Topic**	**E-Safety**	**Python Turtle**	**Computer Systems**	**Small Basic**	**Cybersecurity**	**Algorithms**
	Enquiry Question	How can we behave in a way that is appropriate and respectful online?	How can we use a programming language to create graphics and games?	How does hardware and software determine our use and control of computer systems?	How can we use a programming language to follow/execute instructions?	What techniques do cybercriminals use to steal data, disrupt systems, and infiltrate networks?	How can we use flowcharts to solve complex problems?
	Key Knowledge and skills	• Students can identify spam, phishing and inappropriate content • Know how to set up social media accounts so that they are private Students can be respectful whilst browsing and interacting with others online	• Students can use problem solving skills in code • Students can read and code using a high-level programming language • Students can identify and rectify logic and syntax errors in code	• Students can identify what inputs and outputs are • Students can identify the purpose of primary and secondary storage • Students can identify the roles of each component of computer hardware	• Students can use functions • Students develop problem solving skills • Students develop computational thinking skills	• Students know companies can harvest/misuse personal data • Know how privacy policies work • Students can identify social engineering techniques • Students know hackers can exploit systems Understand the risks of malware	• Know how to use the features of a flowchart – start/stop, process, input and decision • Know how to use branches/nodes to direct flow • Know how to visually represent a problem-solving process
	End Point	Students can identify risks when browsing the internet Students are able to differentiate between email and spam Students are aware of the help available if attacked online	Students can use basic commands and functions in python to create graphics and interactive animations	Students can identify key software and hardware of a computer system Students can make informed decisions about what equipment is best suited for a specific task or scenario	Students can create effective code and programs that perform simple to intermediate tasks	Students can identify the most effective methods to prevent cyberattacks Students know a range of ways to use technology securely, including protecting their online identity and privacy	Students can solve problems by building effective algorithms as flowcharts
Year 9	**Topic**	**E-Safety**	**Python Programming**	**Networks**	**Augmented Reality**	**Processors and Data**	**Ethics & Law**
	Enquiry Question	What are some of the dangers and risks to young adults and children when using the internet?	How can I create programs using a high-level programming language?	What are the 'internet' and the 'World Wide Web', what are the benefits of networks?	How can I explore augmented reality to create unique images and graphics?	How does a computer process different data types?	What are the laws that govern computing?
	Key Knowledge and skills	• Students can identify abuse and explain types • Students can discuss how extremists use the internet to groom • Students create effective websites /multimedia content	• Students can create robust code • Students can document their process • Students can reflect on programming •	• Students describe components of networks and how they work together • Students know the difference between the internet, its services, and the World Wide Web	• Students can overlay digital content onto real-life environments and objects. • Students are able to create graphics which are both interactive and immersive	• Students can identify the main CPU components • Students familiar with data units • Students know storage devices have different capacities • Students know the advantages and disadvantages of compression	• Students can describe features of key laws and acts associated with computing • Students can identify impacts of technology and how these are experienced, negated or adapted to
	End Point	Students can identify the various types of abuse and grooming and explain these through their own website	Students can create robust programmes that imitate real-life scenarios	Students can identify the hardware used in computer networks and explain the benefits of different network types	Students can create digital artefacts for a given audience, with attention to design and usability	Students can identify how a computer processor works Students can make informed decisions about data units, types and file compression	Students can understand the impact of technology on individuals, organisations and the planet through a range of real-world examples
Year 10	**Topic**	Systems architecture	Memory and storage	Computer networks, connections & protocols	Network security	Systems software	Ethical, legal, cultural & environmental impacts of digital technology
	Enquiry Question	What are the main characteristics of computer systems?	What is the purpose of memory and storage in a computer system?	How do different types of networks function?	How can security risks be identified and managed?	How does software work to support a computer system?	What are the potential impacts of digital technology?
	Key Knowledge and skills	**Architecture of the CPU** • Purpose of the CPU • Common CPU components and their function • Von Neumann architecture **CPU Performance** • How common characteristics of CPUs affect performance **Embedded Systems** • The purpose and characteristics of embedded systems • Examples of embedded systems	**Primary Storage** • The need for primary storage • The difference between RAM & ROM • The purpose of ROM in a computer system • The purpose of RAM in a computer system • Virtual memory **Secondary Storage** • The need for secondary storage • Common types of storage • Suitable storage devices and storage media for a given application • The advantages and disadvantages of different storage **Units of Storage** • Units of data storage	**Networks & Topologies** • Types of network • Factors that affect the performance of networks • The different roles of computers in a client-server and a peer-to-peer network • The hardware needed to connect stand-alone computers into a Local Area Networks • The Internet as a worldwide collection of computer networks • Star & Mesh topologies **Wired & Wireless Networks** • Modes of connection • Encryption • IP addressing and MAC addressing	**Threats** • Forms of attack: ○ Malware ○ Social engineering ○ Brute-force attacks ○ Denial of service attacks ○ Data interception and theft ○ The concept of SQL injection **Vulnerabilities** • Common prevention methods: ○ Penetration testing ○ Anti-malware software ○ Firewalls ○ User success levels ○ Passwords ○ Encryption	**Operating Systems** • The purpose and functionality of operating systems **Utility Software** • The purpose and functionality of utility software	**Ethics** • Impacts of digital technology on wider society including: ○ Ethical ○ Legal ○ Cultural ○ Environmental ○ Privacy **Legislation** • Legislation relevant to Computer Science ○ The Data Protection Act 2018 ○ Computer Misuse Act 1990 ○ Copyright Designs and Patents Act 1988 Software licences (i.e. open source and proprietary)

Computing: Years 7, 8 and 9; Computer Science: Years 10 and 11

			● How data is converted into a binary format to be processed ● Data capacity and calculation of data requirements **Data Storage (Numbers)** ● How to add two binary integers together ● How to convert denary whole numbers to binary numbers ● How to convert denary whole numbers into 2-digit hexadecimal numbers & vice versa ● How to convert binary values to hexadecimal & vice versa ● Binary shifts **Data Storage (Character)** **ASCII and Unicode** ● The use of binary codes to represent characters ● Character sets ● The relationship between the number of bits per character in a character set, and the number of characters which can be represented **Data Storage (Image)** ● How an image is represented as a series of pixels, represented in binary ● Metadata ● The effect of colour depth and resolution on the quality of an image & the size of an image Students **Data Storage (Sound)** ● How sound can be sampled and stored in digital form ● The effect of sample rate, duration and bit depth on: playback quality & size of a sound file **Compression** ● The need for compression ● Types of compression	**Protocols & Layers** ● Common protocols: ○ TCP/IP ○ HTTP ○ HTTPS ○ FTP ○ POP ○ IMAP ○ SMTP ● Concept of layers	● Physical security		
	End Point	Students are familiar with the role of each component in the CPU and what actions occur at each stage of the fetch-execute cycle Students know the effects of changes to the performance of the CPU Students know what embedded systems are and are familiar with a range of them	Students know why computers have primary & secondary storage and how virtual memory works Students know why data is stored in binary format, are familiar with data units and are able to calculate data capacities Students understand binary, denary and hexadecimal numbers and can convert between them Students understand how data is represented and stored as characters, images and sound Students know scenarios where compression is used, the advantages and disadvantages of each compression type and the effect on files	Students know the characteristics of networks, common examples of LANs and WANs, including star and mesh topologies Students understand the factors that can affect the performance of a network and the role of hardware Students understand the concept of the internet and how it works Students understand the concept of the cloud, its advantages and disadvantages Students can compare wired and wireless connections and know the principles of encryption Students know the principles and types of communication protocols	Students know how threats are posed to devices and systems Students know how to limit security threats posed and the methods to remove vulnerabilities to devices and systems	Students can explain what each function of an operating system does and the features of a user interface Students understand why data transfer between devices and the processor needs to be managed Students know the key features of user management and file management Students understand how computers use utility software to perform housekeeping tasks such as encryption, defragmentation and data compression	Students can describe the features of key legislation and acts associated with data and computer systems Students understand the need to license software Students can identify the impacts of technology and how these are experienced, negated or adapted to
Year 11	**Topic**	Algorithms	Programming fundamentals	Producing robust programs	Boolean logic	Programming languages and integrated development environments	
	Enquiry Question	How are computational thinking skills used to refine and solve problems?	How can we use SQL to search for data?	How can we use programming techniques to create robust programs?	How can we represent logic in diagrams and tables?	How can we demonstrate an understanding of the key features of an IDE?	
	Key Knowledge and skills	**Computational Thinking** ● Principles of computational thinking ○ Abstraction ○ Decomposition	**Additional Programming Techniques** ● Use of SQL to search for data ○ SELECT	**Extended Programming Project** ● The use of basic string manipulation ● The use of records to store data	**Boolean Logic** ● Simple logic diagrams using the operators AND, OR and NOT	**Languages** ● Characteristics and purpose of different levels of programming language ● The purpose of translators	

PONTEFRACT ACADEMIES TRUST

Computing: Years 7, 8 and 9; Computer Science: Years 10 and 11

		○ Algorithmic Thinking ● Designing, creating and refining algorithms ● Identify the inputs, processes, and outputs for a problem ● Structure diagrams ● Create, interpret, correct, complete, and refine algorithms ● Identify common errors ● Trace tables **Create, interpret, correct, complete and refine algorithms using:** ● Pseudocode ● Flowcharts **Searching and Sorting** ● Standard searching algorithms: ● binary search ● linear search ● Standard sorting algorithms: ● bubble sort ● merge sort • insertion sort	○ FROM WHERE	● The use of arrays (or equivalent) when solving problems, including both one-dimensional and two-dimensional arrays ● How to use subprograms (functions and procedures) to produce structured code • Random number generation	● Truth tables ● Combining Boolean operators using AND, OR and NOT • Applying logical operators in truth tables to solve problems	● The characteristics of a compiler and an interpreter **Integrated Development Environment** ● Common tools and facilities available in an Integrated ● Development Environment (IDE): ● Editors ● Error diagnostics ● Run-time environment Translators	
End Point	Students are familiar with computational thinking principles and how they are used to define / refine problems Students can produce a simple diagram to show the structure of a problem and its subsections Students can create and use trace tables to follow an algorithm Students understand the main steps of search and sort algorithms and can apply an algorithm to a data set Students can identify and algorithm from code or pseudocode	Students understand the practical use of programming techniques in a high-level language Students can recognise and use comparison and arithmetic operators Students can use SQL commands to search for data	Students understand the use of basic string manipulation Students can use records to store data and they can use arrays when solving problems Students know where to use functions and procedures effectively Students can create and use random numbers in a program	Students know the truth tables for each logic gate and can recognise gate symbols Students can create and edit logic diagrams and truth tables	Students understand difference between high and low-level programming languages Students are familiar with the need for translators Students know the differences, benefits and drawbacks of using a complier or an interpreter Students know the tools that an IDE provides Students know how the tools and facilities of an IDE help a programmer to develop a program Students have practical experience of using these tools in an IDE	