	Half term 1	Half term 2	Half term 3	Half term 4	Half term 5	Half term 6
Year 10 Topic	**Systems Architecture** **Primary storage**	**Secondary storage** **Networks -Wired and wireless** **Algorithms**	**Networks** **Topologies, Protocols & Layers** **Algorithms** **Programming Techniques**	**Systems Security & Software** **Producing Robust Programs** **Computational logic**	**Computational logic** **Translators & Facilities**	**Data Representation** **Programming Techniques**
I am learning about:	We learn about the purpose of the CPU • Von Neumann architecture: - MAR (Memory Address Register) - MDR (Memory Data Register) - Program Counter - Accumulator • common CPU components and their function: - ALU (Arithmetic Logic Unit) - CU (Control Unit) - Cache • the function of the CPU as fetch and execute instructions stored in memory • how common characteristics of CPUs affect their performance: - clock speed - cache size - number of cores • embedded systems: - purpose of embedded systems	We learn the need for secondary storage • data capacity and calculation of data capacity requirements • common types of storage: - optical - magnetic - solid state • suitable storage devices and storage media for a given application, and the advantages and disadvantages of these, using characteristics:- capacity - speed - portability - durability - reliability - cost We learn the types of networks: - LAN (Local Area Network) - WAN (Wide Area Network) • factors that affect the performance of networks • the different roles of computers in a client-server	We learn about star and mesh network topologies • Wifi: - frequency and channels - encryption • ethernet • the uses of IP addressing, MAC addressing, and protocols including: - TCP/IP (Transmission Control Protocol/Internet Protocol) - HTTP (Hyper Text Transfer Protocol) - HTTPS (Hyper Text Transfer Protocol Secure) - FTP (File Transfer Protocol) - POP (Post Office Protocol) - IMAP (Internet Message Access Protocol) - SMTP (Simple Mail Transfer Protocol) • the concept of layers • packet switching. We learn the use of variables, constants, operators, inputs, outputs and assignments • the use of the three	We learn about the different forms of attack • threats posed to networks: - malware - phishing - people as the 'weak point' in secure systems (social engineering) - brute force attacks - denial of service attacks - data interception and theft - the concept of SQL injection - poor network policy • identifying and preventing vulnerabilities: - penetration testing - network forensics - network policies - anti-malware software - firewalls - user access levels - passwords - encryption. We learn the • purpose and functionality of systems software • operating systems: - user interface	We learn why data is represented in computer systems in binary form • simple logic diagrams using the operations AND, OR and NOT • truth tables • combining Boolean operators using AND, OR and NOT to two levels • applying logical operators in appropriate truth tables to solve problems • applying computing-related mathematics: - + - – - / - * -Exponentiation (^) -MOD - DIV We learn the characteristics and purpose of different levels of programming language, including low level languages	We will learn about Units • bit, nibble, byte, kilobyte, megabyte, gigabyte, terabyte, petabyte • how data needs to be converted into a binary format to be processed by a computer. Numbers • how to convert positive denary whole numbers (0–255) into 8 bit binary numbers and vice versa • how to add two 8 bit binary integers and explain overflow errors which may occur • binary shifts • how to convert positive denary whole numbers (0–255) into 2 digit hexadecimal numbers and vice versa

	-examples of embedded systems. We learn the difference between RAM and ROM • the purpose of ROM in a computer system • the purpose of RAM in a computer system • the need for virtual memory • flash memory	and a peer-to-peer network • the hardware needed to connect stand-alone computers into a Local Area Network: - wireless access points - routers/switches - NIC (Network Interface Controller/Card) - transmission media • the internet as a worldwide collection of computer networks: - DNS (Domain Name Server) - hosting - the cloud • the concept of virtual networks. We learn about computational thinking: - abstraction - decomposition - algorithmic thinking • standard searching algorithms: - binary search - linear search • standard sorting algorithms: - bubble sort - merge sort - insertion sort • how to produce algorithms using: - pseudocode - using flow diagrams • interpret, correct or complete algorithms.	basic programming constructs used to control the flow of a program: - sequence - selection - iteration (count and condition controlled loops) • the use of basic string manipulation • the use of basic file handling operations: - open - read - write - close • the use of records to store data • the use of SQL to search for data • the use of arrays (or equivalent) when solving problems, including both one and two dimensional arrays • how to use sub programs (functions and procedures) to produce structured code • the use of data types: - integer - real - Boolean - character and string - casting • the common arithmetic operators • the common Boolean operators.	- memory management/multitasking - peripheral management and drivers - user management - file management • utility system software: - encryption software - defragmentation - data compression - the role and methods of backup: •full • incremental. We learn defensive design considerations: - input sanitisation/validation - planning for contingencies - anticipating misuse - authentication • maintainability: - comments - indentation • the purpose of testing • types of testing: -Iterative - final/terminal • how to identify syntax and logic errors • selecting and using suitable test data.	• the purpose of translators • the characteristics of an assembler, a compiler and an interpreter • common tools and facilities available in an integrated development environment (IDE): -editors -error diagnostics -run-time environment - translators.	• how to convert from binary to hexadecimal equivalents and vice versa • check digits. Characters • the use of binary codes to represent characters • the term 'character-set' • the relationship between the number of bits per character in a character set and the number of characters which can be represented (for example ASCII, extended ASCII and Unicode). Images • how an image is represented as a series of pixels represented in binary • metadata included in the file • the effect of colour depth and resolution on the size of an image file. Sound • how sound can be sampled and stored in digital form • how sampling intervals and other factors affect the size of a sound file

						and the quality of its playback: - sample size - bit rate - sampling frequency. Compression • need for compression • types of compression: - lossy - lossless
Assessment	Assignments on Seneca, Google Classroom, Bitesize	Assignments on Seneca, Google Classroom, Bitesize	Assignments on Seneca, Google Classroom, Bitesize	Assignments on Seneca, Google Classroom, Bitesize	Assignments on Seneca, Google Classroom, Bitesize	Assignments on Seneca, Google Classroom, Bitesize
Year 11 Topic	**Recap COMP 1 Topics** **Programming project**	**Ethical, legal, cultural and environmental concerns** **Programming project & Flowcharts/algorithms**	**Programming project & Problem Techniques Data Representation** **Problem solving using pseudo code**	**Programming project & Problem Solving Translators & Facilities** **Computational logic**	**Revision/ Past Papers**	**Revision / Past Papers**
I am learning about:	We learn how to identify and use variables, operators, inputs, outputs and assignments • how to understand and use the three basic programming constructs used to control the flow of a program: Sequence; Selection; Iteration • how to understand and use suitable loops including count and condition controlled loops • how to use different types of data, including	We learn how to investigate and discuss Computer Science technologies while considering: - ethical issues - legal issues - cultural issues - environmental issues. - privacy issues. • how key stakeholders are affected by technologies • environmental impact of Computer Science • cultural implications of Computer Science	We learn how to design suitable algorithms to represent the solution to a problem • how to design suitable input and output formats and navigation methods for their system • how to identify suitable variables and structures with appropriate validation for their system • how to use appropriate data types in their system • how to use	Learn how how to develop a solution to the identified problem using a suitable programming language(s) • how to demonstrate testing and refinement of the code during development • how to explain the solution using suitable annotation and evidence of development • how to use suitable techniques to solve all aspects of the problem	Revise all topics for both COMP 1 paper and COMP 2 paper	Revise all topics for both COMP 1 paper and COMP 2 paper

	Boolean, string, integer and real, appropriately in solutions to problems • how to understand and use basic string manipulation • how to understand and use basic file handling operations: - open - read - write - close • how to define and use arrays (or equivalent) as appropriate when solving problems • how to understand and use functions/sub programs to create structured code. We will learn how to analyse and identify the requirements for a solution to the problem • how to set clear objectives that show an awareness of the need for real world utility • how to use abstraction and decomposition to design the solution to a problem • how to identify the data requirements for their system • how to identify test procedures to be used during and after development to check their system against the success criteria	• open source vs proprietary software • legislation relevant to Computer Science: - The Data Protection Act 1998 - Computer Misuse Act 1990 -Copyright Designs and Patents Act 1988 - Creative Commons Licensing - Freedom of Information Act 2000 Learn about Flow charts like pseudocode are informal but the most common flow chart shapes are: Line An arrow represents control passing between the connected shapes. Process This shape represents something being performed or done. Sub Routine This shape represents a subroutine call that will relate to a separate, non-linked flow chart Input/Output This shape represents the input or output of something into or out of the flow chart. Decision This shape represents a decision (Yes/No or True/False) that results in two lines representing the different possible outcomes. Terminal This shape represents the “Start” and “End” of the process.	functions/sub programmes to produce structured reusable code • how to select suitable techniques for the development of the solution. Learn how Variables and constants are assigned using the = operator. x=3 name=”Bob” Variables and constants are declared the first time a value is assigned. They assume the data type of the value they are given. Variables and constants that are declared inside a function or procedure are local to that subroutine. Variables in the main program can be made global with the keyword global. global userid = 123 Variables in the main program can be made constant with the keyword const. const vat = 20 Revise how casting Variables can be typecast using the int str and float functions. str(3) returns “3” int(“3”) returns 3 float(“3.14”) returns 3.14 Outputting to screen print(string) print(variable) Example print(“hello”) print(myAge) Taking Input from User variable=input(prompt to	• how to take a systematic approach to problem solving • how to deploy practical techniques in an efficient and logical manner • how to show an understanding of the relevant information by presenting evidence of the development of their solutions • how to show an understanding of the technical terminology/concepts that arise from their investigation through analysis of the data collected • how to use the terminology/concepts surrounding their topic and contained in the information collected correctly when it comes to producing analysis in the supporting script. We learn how to produce a full report covering all aspects of the investigation • how to present the information in a clear form which is understandable by a third party and which is easily navigatable • how to critically appraise the evidence that they have presented • how to test their own solution		

	• how to use validation to ensure a robust solution to a problem.		user) Example name=input("Please enter your name")	• how to present their evaluation in a relevant, clear, organised, structured and coherent format • how to use specialist terms correctly and appropriately • how to present a conclusion to the report • how to justify their conclusions based on the evidence provided.		
Assessment	Assignments on Seneca, Google Classroom, Bitesize, programming project	Assignments on Seneca, Google Classroom, Bitesize, programming project	Assignments on Seneca, Google Classroom, Bitesize	Assignments on Seneca, Google Classroom, Bitesize	Past papers	**Examination** paper 1 and 2