The King's CHURCH OF ENGLAND ACADEMY KIDSGROVE

Curriculum plan: Computing

CONNECTED

INTENT:

"I do not fear computers, I fear the lack of them"

Isaac Asimov

The use and understanding of computers gives learners the opportunity to develop sector-specific knowledge and skills in a practical learning environment. Can you solve problems? Are you a computational thinker? Do you have a scientific and mathematical way of thinking?

Computers are changing every part of our lives at an ever-increasing rate-why not drive the future?

The King's CHURCH OF ENGLAND ACADEMY KIDSGROVE

Curriculum plan: Computing

CONNECTED

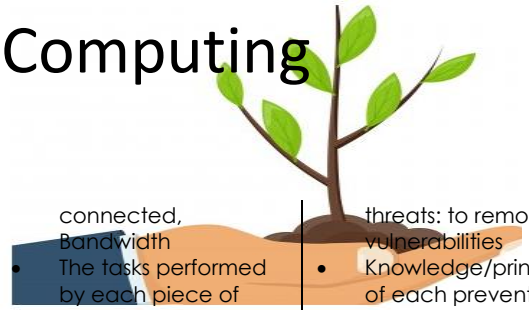
Please click on the icons to access our online portal where you can learn more about each topic*

Half term points

	AUTUMN 1	AUTUMN 2	SPRING 1	SPRING 2	SUMMER 1	SUMMER 2
	Systems architecture	**Memory and storage**	**Computer networks, connections and protocols**	**Network security**	**Systems software**	**Ethical, legal, cultural and environmental impacts of digital technology**
10	**Learning to include:** • Architecture of the CPU including: The purpose of the CPU, common CPU components/functions and the Von Neuman architecture. • CPU performance including: clock speed, cache size and number of cores. • Embedded Systems including: characteristics of embedded • What actions occur at each stage of the fetch-execute cycle • The role/purpose of each component and what it manages, stores, or controls during the fetch-execute cycle	**Learning to include:** • Primary storage (memory) to include RAM and ROM, the need for primary storage • Secondary storage to include: common types of storage, suitable storage devices. Advantages and disadvantages of storage devices. • Units of storage, numbers and characters. • Compression: lossy and lossless. • Why computers have primary and secondary storage • Key characteristics of RAM and ROM • Why virtual memory may be needed in a system	**Learning to include:** • Types of networks (LAN and WAN) • Factors affecting performance. • Hardware and software. • Network Topologies (STAR, MESH, BUS) • Wired vs Wireless networks. • Encryption and IP addresses. • Common Protocols: HTTPS, IMAP, SMTP, POP. • The characteristics of LANs and WANs including common examples of each • Understanding of different factors that can affect the performance of a network, e.g.: Number of devices	**Learning to include:** • Threat to a computer systems including: Malware, social engineering, brute-force attacks, DNS attacks, Data interception and SQL injection. • Preventing vulnerabilities including: anti-malware, firewalls, User access levels, Encryption, Physical security. • Threats posed to devices/systems Knowledge/principles of each form of attack including: How the attack is used, the purpose of the attack • Understanding of how to limit the	**Learning to include:** • Operating systems: user interface, memory management and multitasking, peripheral management and driver. • Utility software including: encryption, defragmentation, compression • What each function of an operating system does • Features of a user interface • Memory management, e.g. the transfer of data between memory, and how this allows for multitasking • Understand that: • Data is transferred between devices and the processor	**Learning to include:** • Impacts of digital technology on wider society including: Ethical issues, Legal issues, Cultural issues, Environmental issues, Privacy issues • 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) • Technology introduces ethical, legal, cultural, environmental and privacy issues • Knowledge of a variety of examples of digital technology and how this impacts on society

The King's CHURCH OF ENGLAND ACADEMY KIDSGROVE

Curriculum plan: Computing



CONNECTED

	• The purpose of each register, what it stores (data or address) • The difference between storing data and an address • Understanding of each characteristic as listed • The effects of changing any of the common characteristics on system performance, either individually or in combination • Typical characteristics of embedded systems • Familiarity with a range of different embedded systems Teach-ICT For Computer Science boqeszp Craig n' Dave Khan Academy	• Differences between each type of storage device/medium • Why data must be stored in binary format • Calculate file sizes of sound, images and text files • Denary , Hex and Binary number systems • How characters are represented in binary and character set codes • Analogue sounds must be stored in binary Teach-ICT For Computer Science boqeszp Craig n' Dave Khan Academy	connected, Bandwidth • The tasks performed by each piece of hardware • The concept of the Internet as a network of computer networks • The Cloud: remote service provision (e.g. storage, software, processing), Advantages and disadvantages of the Cloud, MESH, Star topologies, wired and wireless. • The principle of encryption to secure data across network connections Teach-ICT For Computer Science boqeszp Craig n' Dave Khan Academy	threats: to remove vulnerabilities • Knowledge/principles of each prevention method: • What each prevention method may limit/prevent • How it limits the attack Teach-ICT For Computer Science boqeszp Craig n' Dave Khan Academy	• This process needs to be managed • User management functions, e.g.: Allocation of an account, Access rights, Security, etc. • File management, and the key features, e.g.: Naming, Allocating to folders , Moving files, Saving, etc. • Understand that computers often come with utility software, and how this performs housekeeping tasks • Purpose of the identified utility software and why it is required Teach-ICT For Computer Science boqeszp Craig n' Dave Khan Academy	• An ability to discuss the impact of technology based around the issues listed • The purpose of each piece of legislation and the specific actions it allows or prohibits • The need to license software and the purpose of a software licence • Features of open source (providing access to the source code and the ability to change the software) • Features of proprietary (no access to the source code, purchased commonly as off-the-shelf) • Recommend a type of licence for a given scenario including benefits and drawbacks Teach-ICT For Computer Science boqeszp Craig n' Dave Khan Academy

The King's CHURCH OF ENGLAND ACADEMY KIDSGROVE

Curriculum plan: Computing

CONNECTED

Please click on the icons to access our online portal where you can learn more about each topic

Half term points

	AUTUMN 1	AUTUMN 2	SPRING 1	SPRING 2	SUMMER 1	SUMMER 2
11	**Algorithms** **Learning to include:** • Computational thinking: abstraction, decomposition, Algorithmic thinking. • Designing, creating and refining algorithms. – Inputs-process-outputs. • Create, pseudocode, flowcharts and use high level programming languages. • Trace testing, identify common errors. • Searching and sorting algorithms: binary search, linear search, bubble sort, merge sort, insertion sort. • Understanding of these principles and how they are used to define and refine problems • Produce simple diagrams to show: The structure of a problem.	**Programming fundamentals** **Learning to include:** • The use of variables, constants, operators, inputs, outputs and assignments • The use of the three basic programming constructs used to control the flow of a program: Sequence, Selection and Iteration (count- and condition-controlled loops). • The common arithmetic operators. Boolean operators AND, OR and NOT. • The use of data types. • The use of basic string manipulation • The use of basic file handling operations. • Practical use of the techniques in a high-level language within the classroom	**Producing robust programs** **Learning to include:** • Defensive design considerations: Anticipating misuse, Authentication, Input validation, Maintainability. • The purpose of testing • Types of testing: Iterative, Final/terminal • Identify syntax and logic errors • Selecting and using suitable test data: Normal, Boundary, Invalid/Erroneous • Refining algorithms • Understanding of the issues a programmer should consider to ensure that a program caters for all likely input values • Understanding of how to deal with invalid data in a program,	**Boolean logic** **Learning to include:** • Simple logic diagrams using the operators AND, OR and NOT • Truth tables • Combining Boolean operators using AND, OR and NOT • Applying logical operators in truth tables to solve problems . • Knowledge of the truth tables for each logic gate • Recognition of each gate symbol • Understanding of how to create, complete or edit logic diagrams and truth tables for given scenarios • Ability to work with more than one gate in a logic diagram	**Programming languages and Integrated Development Environments** **Learning to include**: • Characteristics and purpose of different levels of programming language: High-level languages, Low-level languages • The purpose of translators • The characteristics of a compiler and an interpreter • Common tools and facilities available in an Integrated Development Environment (IDE): Editors, Error diagnostics, Run-time environment, Translators • The differences between high- and low-level programming languages • The need for translators	**Exam revision** **Learning to include:** J277/01: Computer systems • This component will assess: 1.1 Systems architecture, 1.2 Memory and storage, 1.3 Computer networks, connections and protocols, 1.4 Network security , 1.5 Systems software, 1.6 Ethical, legal, cultural and environmental impacts of digital technology J277/02: Computational thinking, algorithms and programming • This component will assess: 2.1 Algorithms, 2.2 Programming fundamentals, 2.3 Producing robust programs, 2.4 Boolean logic, 2.5

The King's CHURCH OF ENGLAND ACADEMY KIDSGROVE

Curriculum plan: Computing

CONNECTED

	Subsections and their links to other subsections • Complete, write or refine an algorithm using the techniques listed • Identify syntax/logic errors in code and suggest fixes • Create and use trace tables to follow an algorithm . • Understand the main steps of each algorithm • Understand any pre-requisites of an algorithm • Apply the algorithm to a data set • Identify an algorithm if given the code or pseudocode for it Teach-ICT For Computer Science boqeszp Craig n' Dave Khan Academy	• Understanding of each technique • Practical use of the data types in a high-level language within the classroom • Ability to choose suitable data types for data in a given scenario • Understand that data types may be temporarily changed through casting, and where this may be useful • Practical use of the additional programming techniques in a high-level language within the classroom Teach-ICT For Computer Science boqeszp Craig n' Dave Khan Academy	Authentication to confirm the identity of a user • Practical experience of designing input validation and simple authentication • Understand why commenting is useful and apply this appropriately • The difference between testing modules of a program during development and testing the program at the end of production • Understand Syntax/logic errors • Types of test data: Normal, Boundary and Erroneous test Teach-ICT For Computer Science boqeszp Craig n' Dave Khan Academy	Teach-ICT For Computer Science boqeszp Craig n' Dave Khan Academy	• The differences, benefits and drawbacks of using a compiler or an interpreter • Knowledge of the tools that an IDE provides • How each of the tools and facilities listed can be used to help a programmer develop a program • Practical experience of using a range of these tools within at least one IDE Teach-ICT For Computer Science boqeszp Craig n' Dave Khan Academy	Programming languages and Integrated Development Environments Teach-ICT For Computer Science boqeszp Craig n' Dave Khan Academy