Computing

Curriculum overview

All children are entitled to a curriculum and to the powerful knowledge which will open doors and maximise their life chances. Below is a high-level overview of the critical knowledge children will learn in this particular subject, at each key stage from Reception through to Year 11, in order to equip students with the cultural capital they need to succeed in life. The curriculum is planned vertically and horizontally giving thought to the optimum knowledge sequence for building secure schema.

	Knowledge; skills and understanding to be gained at each stage*		
	Cycle 1	Cycle 2	Cycle 3
YEAR 1	**Safe use of technology** Logging on and setting a secure password **Use of technology** Technology outside of school (self-service checkouts / ticket machines); impact to society	**Personal Information** How to keep safe online **Careers in ICT / Computing** Common careers in the field and routes to get there	**Algorithms** Importance of clear instructions within a computer system **Scratch** Directional programming in a block-based environment
YEAR 2	**Creating programs** Planning and implementation of own programs **Debugging programs** Looking for errors in code	**Logical reasoning** Using patterns and logic to solve problems **Algorithm prediction** Tracing pre-designed algorithms in order to prodict their outcomes	**Creating digital content** Creative freedom to produce a web blog, digital graphic or video surrounding algorithms **Storing / editing and organising digital content** Enhancing the aesthetics and creating folders for organisation
YEAR 3	**Designing, writing and debugging algorithms** What are algorithms and how do we write them? Use of flowchart shapes to create solution; coding in scratch **Physical systems** Creating working systems with hardware or through online software	**Decomposition** Investigating how to break down a large problem into smaller, independent programs **Abstraction** Removal of the non-essential data so as to focus on important elements	**Sequence, selection, iteration** Using options within their programs and investigate the best way to structure code Use of loops and evaluate the effectiveness on the efficiency
YEAR 4	**Variables** Identify the purpose and funcxtionality of variables in programs. Investigate: Why are variables used in programming? **Hardware and software** Students will investigate the hardware needed to build a computer systems and the role of software	**Input and output systems** Research different devices and their purpose within a computer **Storage devices** Students will investigate how data is stored on a computer system and research both primary and secondary storage methods	**Memory** Explore how data is stored in binary and how devices communicate within a computerized system **Coding** Block based coding (scratch / Kodu) using IF / THEN statements
YEAR 5	**Debugging** Investigating errors in code and writing solutions **Networking** LAN, MAN, WAN, peer to peer networks How devices communicate in a network	**Internet** Explore the WWW and internet as a collection of web pages **Communication and collaboration** Investigate how devices can be used as a communication and collaboration tool	**Keeping safe online** Investigating methods of keeping information safe, using websites correctly and reporting concerns **Analysing digital content** Assessing against a brief and identifying improvements
YEAR 6	**Sensible use of technology** Exploring ways to use websites safely **Software for presenting** Use of PowerPoint and Publisher to produce presentational pieces	**Producing digital content** Reviewing online content and creating own web blog / website **Games design** Producing own games in Scratch / alternative block-based coding	**Games development** Producing games in scratch using variables, iteration, sequence and selection **Testing and evaluation** Writing accurate tests to assess the functionality of developed game Recommendations for future development
YEAR 7	**E-safety and digital literacy** Recognise reliable sources of information and list ways of keeping safe online	**Algorithms** Importance of writing clear instructions	**Python programming** Writing programs in Python using sequence, selection and iteration

	Knowledge; skills and understanding to be gained at each stage*		
	Cycle 1	**Cycle 2**	**Cycle 3**
YEAR 7	**Hardware and software** Investigate the hardware needed to build a PC system and the software that runs on top	**Computational thinking** Thinking like a computer in order to solve a problem	**Binary** Converting from binary to denary and visa versa Performing binary addition and explaining the overflow error
YEAR 8	**Operating systems and security** Investigating mobile and PC Operating systems **Python programming** Introduction to programming using driver / navigator roles	**Binary and hexadecimal** Converting from binary to denary and visa versa Performing binary addition and explaining the overflow error **CPU** Investigating the role of the central processing unit and the ALU, CU and cache	**Gaming constructs** Investigate how a game is made and the different components of a game **Cyber security** Investigate security threats to networks and how attacks are carried out
YEAR 9	**Pre-production (animation)** Types and techniques used in animation History of animation Planning and designing (mindmaps / moodboards / script / storyboard / assets + resources) **Production (animation)** Using plan to create animation in Animate **Post-production (animation)** Exporting and reviewing animation Analysing against the brief; recommendations for future developments	**Hardware and software** Investigate the hardware needed to build a PC system and the software that runs on top Evidence the way hardware components communicate, including the FDE cycle **Memory** Primary and secondary storage, including RAM and cache **Operating Systems** Function of operating systems and the role of OS security	**Programming (selection)** Writing in Python using variables, inputs processes and outputs **Programming (sequence)** Deciding upon the most effective structure for programming **Programming (iteration)** Implementing loops into programs and evaluate the efficiency
YEAR 10	**OCR GCSE Computer Science** **Component 1 – systems architecure** Memory; storage; wired and wireless networks; network topologies; protocols and layers; system security; system software; ethical, legal, cultural and environmental concerns	**OCR GCSE Computer Science** **Component 2 – algorithms and programming** Algorithms; programming techniques; producing robust programs; computational logic; translators and facilities of languages; data representation	**OCR GCSE Computer Science** **Component 3 – project / exam prep** Programming techniques; analysis; design; development; testing, evaluation and conclusions
	OCR Creative iMedia **R081 LO1** Mood boards; mind maps / spider diagrams; visualisation diagrams; storyboards; scripts Camera shots / angles; movements; lighting; sounds; locations **LO2** Target audience; work plan / production schedule; hardware / software; resources; legislation **LO3** File formats; creating / editing resources; naming conventions; legislation **LO4** Review a pre-production document against a given brief Identify areas of improvement in a pre-production document	**OCR Creative iMedia** **R084 LO1** Why and how Comic strips are used, the origin of comic strips, formats, properties, how purposes and audiences influence the design and layout. **LO2** Interpret client requirements for a multipage comic strip, understand the target audience and the impact this will have on the product, produce a work plan, visualisation diagram, identify assets and resources, legislation. Create a storyboard. **LO3** How to source / create / edit and export storyboards using a range of tools and techniques and version control **LO4** Review storyboard against a brief and identify areas for improvement	**OCR Creative iMedia** **R086 LO1** The purpose and uses of animation, animation types and techniques, file formats, properties, how purposes and audiences influence the design and layout **LO2** Interpret client requirements for a digital graphic, understand the target audience and the impact this will have on the product, produce a work plan, visualisation diagram, identify assets and resources, legislation, storyboard **LO3** How to source/create/edit and export animation using a range of tools and techniques and version control **LO4** Review animation against a brief and identify areas for improvement
YEAR 11	**OCR GCSE Computer Science** **Component 1** Systems architecure; recap of systems architecure; memory; storage; wired and wireless networks; network topologies; protocols and layers; system security; system software; ethical, legal, cultural and environmental concerns	**OCR GCSE Computer Science** **Component 2** Algorithms and programming Producing robust programs; computational logic; translators and facilities of languages; data representation	

	Knowledge; skills and understanding to be gained at each stage*		
	Cycle 1	Cycle 2	Cycle 3
YEAR 11	**OCR Creative IMedia** **R082 LO1** Why and how digital graphics are used; the types of digital graphic; file formats; properties; how purposes and audiences influence the design and layout **LO2** Interpret client requirements for a digital graphic; understand the target audience and the impact this will have on the product; produce a work plan; visualisation diagram; identify assets and resources; legislation **LO3** How to source / create / edit and export digital graphics using a range of tools and techniques and version control **LO4** Review a graphic against a brief and identify areas for improvement	**OCR Creative IMedia** Recap of all skills covered and examination re-sit	

*A powerful, knowledge-rich curriculum teaches both **declarative knowledge** (facts; knowing that something is the case; what we think about) and non-declarative or **procedural knowledge** (skills and processes; knowing how to do something; what we think with). There are no skills without bodies of knowledge to underpin them.

In some subjects, a further distinction can be made between substantive knowledge (the domain specific knowledge accrued e.g. knowledge of the past) and disciplinary knowledge (how the knowledge is accrued e.g. historical reasoning).

Please refer to the DAT Curriculum Principles, published on our website, for further information about how we have designed our all-through curriculum.