Computing & ICT

Curriculum Information, Intent and Map

Hutton Church of England Grammar School

Staff:

- Mr N Hulme: **Subject Lead**
- Mrs H Allan
- Mr D Dwyer
- Mrs L Downton

Intent:

The curriculum we offer is designed to prepare our students for life outside of the classroom or for further study in our subject, by exposing them to real life or workplace problems. This is embodied in Timothy 3:17:

"That the man of God may be competent, equipped for every good work"

The Department's Computing & ICT curriculum enable them to become decent, honest young adults who are able to make a positive contribution to their community, whether that be friends, family or the wider world. Through the development of a student's technical knowledge they are able to pass on their understanding to those who need it and use it to help those who may be struggling in a fast developing technological world. It also ensures that each student who passes through the department has the necessary technical skills to enter into the world of employment or further study.

The Computing & ICT Department curriculum therefore covers three broad areas:

- **Digital literacy**: appropriate and responsible use of technology which includes online safety. The use of the Microsoft Office Suite of programs is taught discretely throughout the curriculum.

- **Principles of computing**: computational thinking, programming skills in different programming languages, understanding how computers work. Learning how to use computers to solve problems

- **Information technology**: using ICT to analyse and solve problems and communicate and share ideas with others

All aspects of the curriculum are delivered in a way that relates to real life situations and problems to deepen the understanding of the student as a whole.

Resilience, determination and problem solving skills should be evident in the students that experience and are engaged in our curriculum, skills which are reinforced in all aspects of school life and embodied in Colossians 3:23:

"Whatever you do work at it with all your heart"

Computing Programmes of Study: Key Stages 3 & 4 National Curriculum in England

Purpose of Study

A high-quality computing education equips pupils to use computational thinking and creativity to understand and change the world. Computing has deep links with mathematics, science, and design and technology, and provides insights into both natural and artificial systems. The core of computing is computer science, in which pupils are taught the principles of information and computation, how digital systems work, and how to put this knowledge to use through programming. Building on this knowledge and understanding, pupils are equipped to use information technology to create programs, systems and a range of content. Computing also ensures that pupils become digitally literate – able to use, and express themselves and develop their ideas through, information and communication technology – at a level suitable for the future workplace and as active participants in a digital world.

Aims

The national curriculum for computing aims to ensure that all pupils:

♣ can understand and apply the fundamental principles and concepts of computer science, including abstraction, logic, algorithms and data representation

♣ can analyse problems in computational terms, and have repeated practical experience of writing computer programs in order to solve such problems

♣ can evaluate and apply information technology, including new or unfamiliar technologies, analytically to solve problems

♣ are responsible, competent, confident and creative users of information and communication technology.

Attainment Targets

By the end of each key stage, pupils are expected to know, apply and understand the matters, skills and processes specified in the relevant programme of study

Subject Content in Key Stage 3 & 4

Key Stage 3

Pupils should be taught to:

♣ design, use and evaluate computational abstractions that model the state and behaviour of real-world problems and physical systems

♣ understand several key algorithms that reflect computational thinking; use logical reasoning to compare the utility of alternative algorithms for the same problem

♣ use two or more programming languages, at least one of which is textual, to solve a variety of computational problems; make appropriate use of data structures; design and develop modular programs that use procedures or functions

♣ understand simple Boolean logic and some of its uses in circuits and programming; understand how numbers can be represented in binary, and be able to carry out simple operations on binary numbers

♣ understand the hardware and software components that make up computer systems, and how they communicate with one another and with other systems

♣ understand how instructions are stored and executed within a computer system; understand how data of various types (including text, sounds and pictures) can be represented and manipulated digitally, in the form of binary digits

♣ undertake creative projects that involve selecting, using, and combining multiple applications, preferably across a range of devices, to achieve challenging goals, including collecting and analysing data and meeting the needs of known users

♣ create, re-use, revise and re-purpose digital artefacts for a given audience, with attention to trustworthiness, design and usability

♣ understand a range of ways to use technology safely, respectfully, responsibly and securely, including protecting their online identity and privacy; recognise inappropriate content, contact and conduct and know how to report concerns.

Key Stage 4

All pupils must have the opportunity to study aspects of information technology and computer science at sufficient depth to allow them to progress to higher levels of study or to a professional career.

All pupils should be taught to:

♣ develop their capability, creativity and knowledge in computer science, digital media and information technology

♣ develop and apply their analytic, problem-solving, design, and computational thinking skills

♣ understand how changes in technology affect safety, including new ways to protect their online privacy and identity, and how to identify and report a range of concerns.

Curriculum Map:

Year	Half term 1	Half term 2		Half term 3	Half term 4		Half term 5	Half Term 6

7	Computing baseline test. Commence Esafety unit theory and discussion with written test	E-Safety meets multimedia Creating eSafety video telling Y6 pupils about an aspect of safety. (Moviemaker)		Solving problems like a computer (computational thinking) - Flowol	Computer Control programming using logo, Scratch and Python Turtle		Spreadsheets (gangsta zoo) Formulas and data types	Databases or BBC Microbits
8	Online Identity – Digital footprint – theory unit with online self-marking test and 3 longer answer written questions.	Digital Footprint – create a game or animation in Scratch which informs Y7 students how to protect their digital footprint.		Programming using Python Programming challenge assessment	Graphics Unit – Image editing and theory (Vector and Bitmap graphics)		Web Creation including HTML	Create a website
9	Trends in computing Students research to find information for a timeline of developments in electronic gaming. They create a website to present their findings.	Programming (Python) Assessment: Task 1 and 2 of GCSE NEA programming challenge		Programming / extra lessons from previous topic.	Hardware, Software and logic		Image Editing skills	Website building skills
We follow the OCR Specification J277.								

10	1.1 Systems Architecture 1.2 Memory 2.1 Algorithms	1.3 Storage 2.1 Algorithms 2.2 Programming Techniques 2.3 Producing Robust Programs 2.4 Computational Logic		1.7 Systems Software 1.4 Wired & Wireless Networks 2.2 Programming Techniques 2.3 Producing Robust Programs	2.6 Data Representation Programming Development		NEA	NEA
11	1.5 Network Topologies, Protocols & Layers Coding Challenges	2.5 Translators & Facilities Coding Challenges		1.6 Systems Security 1.8 Issues	1.8 Issues Exam preparation		Exam Preparation	GCSE Examinations
We follow the AQA A level specification								
12	Data Types Programming concepts Number Systems Number Base "Bits and bytes Units"	Constants and variables in a programming language Numbers with a fractional part Character form of a decimal digit ASCII and Unicode		Musical Instrument Digital Interface (MIDI) Returning a value/values from a subroutine Local variables in subroutines Data Compression	"Abstraction Information hiding" "Procedural Abstraction Functional Abstraction" "Data Abstraction		Network Topology Types of Networking between hosts Wireless Networking Relationship between hardware and software	Internal hardware components of a computer The meaning of the stored program concept The processor and its components

	Arithmetic operations in a programming language Relational operations in a programming language Unsigned Binary Arithmetic Signed Binary using two's compliment Boolean Operations in a programming language sound and other data	Constants and variables in a programming language String handling operations in a programming language Error Checking and correction Bit patterns, images, Random number generation in a programming language Exception Handling Analogue / Digital conversion Bitmapped Graphics Digital Representation of Sound Subroutines (procedures/ functions)		Encryption Global variables in a programming language Procedural - oriented programming Encryption Problem Solving Data Structures Single and Multi-Dimensional arrays Following and writing algorithms	Problem Abstraction/ Reduction" Single and Multi-Dimensional arrays Fields records and files "Decomposition Compositio Automaton Finite State Machines Fields records and files Analysis Design Implementation Testing Evaluation Communication Methods Communication Basics		Classification of software System Software Role of an operating system(OS) Classificaton of programming languages Types of program translator Logic Gates Using Boolean Algebra Internal hardware components of a computer	The Fetch-Execute cycle and the role of registers within it The processor instruction set Addressing Modes Machine-code/assembly language operations Factors affecting processor performance Input and output devices Secondary storage devices Individual (moral), social (ethical), legal and cultural issues and opportunities

		Parameters of sub routines Digital						
13	NEA Trees Hash Tables Dictionaries Vectors Graph Traversal Tree Traversal Reverse Polish Searching and sorting algorithms Dijkstra's Shortest Path Algorithm Numbers with a fractional part	NEA Maths for regular expressions "Regular expressions Regular language" Backus-Naur Form ((BNF)/syntax diagrams Analogue - digital conversion Graphics Databases and SQL "Comparing Algorithms		NEA "The Internet and how it works Internet security" "TCP/IP Standard application layer protocols" "IP address structure Subnet masking IP standards Public and private IP addresses"	NEA Exam Preparation		Exam Preparation	A Level Examinations

	"Rounding Errors Absolute and Relative Errors Range and Precision" "Normalisation Underflow and Overflow" Checksums Role of stack frames in subroutine calls Programming Techniques	Maths for understanding Big O notation" "Order of complexity Limits of computation" "Classification of algorithmic problems Computable and non-computable problems" Turing Machine Halting Problem		"Dynamic Host Configuration Protocol (DHCP) Network Address Translation (NAT) Port forwarding" "Client Server Model Thin- versus thick-client computing" Function Type First-class object Function Application Logic Gates Big Data Exam Preparation				

For additional course & curricular information please see:

GCSE: Options Booklet

A Level: Sixth Form Course Information