Curriculum Overview

Computing

Winifred Holtby Academy

Respect, Ambition, Achievement

Introduction

This document outlines the curriculum and key considerations including:

- Aims and purpose
- Alignment with the whole school provision and curriculum intent
- A summary programme of study which includes sequencing of taught content

We use the National Curriculum as our statutory foundation and broadly share its principles and aims including:

- 'To provide students with an introduction to the essential knowledge that they need to be educated citizens. To introduce students to the best that has been thought and said; and help engender an appreciation of human creativity and achievement'.
- To prepare students to be confident in themselves, to have a fulfilled and successful life beyond our school – one where they contribute positively to society.
- Our statutory curriculum is just one element in the education of every child. There is time and space in the school day and in each week, term and year to range beyond statutory specifications.
- Provision of a framework of core knowledge around which teachers can develop exciting and stimulating lessons to promote the development of students' knowledge, understanding and skills as part of the wider school curriculum.
- The wider school curriculum includes an extensive range of opportunities and activities that are routinely available to students, are inclusive and reflect our diverse community.

Numeracy and literacy

Teachers should take opportunities to develop students' mathematical fluency, spoken language, reading, writing and vocabulary within their specific discipline and in line with the expectations laid out in our school curriculum statement.

Purpose of study

'A high-quality computing education equips students 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 students 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, students are equipped to use information technology to create programs, systems and a range of content. Computing also ensures that students 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.' Adapted from National Curriculum, DfE, 2014.

Curriculum Aims

Our Computing curriculum at Key Stage 3 and 4 is designed to instil in students a sense of belief in their own abilities and personal characteristics. Through engaging and challenging activities, students will develop resilience, problem-solving skills, and a growth mindset necessary to thrive in the dynamic field of computing. We aim to foster a culture of curiosity, creativity, and collaboration, where students are encouraged to take risks, learn from failure, and persevere in the face of challenges.

At the core of our curriculum is the belief that every student has the potential to achieve great things. Through rigorous instruction and hands-on experiences, we strive to equip our students with the knowledge, skills, and confidence to shape positive futures for themselves. By providing a rich and diverse learning environment, we aim to prepare students for further study in computing or for successful careers in the digital age. Our curriculum ensures that students develop a deep understanding of fundamental principles and concepts while also honing their analytical, problem-solving, and computational thinking skills.

Our goal is to empower students to succeed not only academically but also as responsible, competent, and creative users of information and communication technology. Our curriculum emphasises the importance of digital literacy, ethical computing practices, and online safety, ensuring that students are equipped to navigate the complexities of the digital world with confidence and integrity.

Building on prior learning

By the end of Key Stage 2, students should have been taught to:

- design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts
- use sequence, selection, and repetition in programs, work with variables and various forms of input and output
- use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs
- understand computer networks, including the internet; how they can provide multiple services, such as the World Wide Web, and the opportunities they offer for communication and collaboration
- use search technologies effectively, appreciate how results are selected and ranked, and be discerning in evaluating digital content
- select, use, and combine a variety of software (including internet services) on a range of digital devices to design and create a range of programs, systems and content that accomplish given goals, including collecting, analysing, evaluating, and presenting data and information
- use technology safely, respectfully, and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact

What are the skills gaps?

Generally, students from feeder primary schools are familiar with using some programmes from Microsoft Office, particularly Word and PowerPoint. Typically keyboard and mouse skills increasingly becoming a weakness due to use of smart devices with touchscreens – this leads to some students lacking facility with keyboards and basic computer skills, file management in particular. Although the primary curriculum covers coding, this is not always taught consistently across all schools due to limitations on resources in some feeder primary schools, so some students do not have an understanding or knowledge of basic coding concepts and skills.

Curriculum Threads (Disciplinary Knowledge)

Disciplinary Knowledge Strands	KS3	KS4 GCSE Computer Science
E Safety	Students develop safe and responsible online behaviours, including protecting personal information, recognising inappropriate content and understanding how to report concerns. As their independence online increases, students explore issues such as misinformation, fake news, digital footprints and the long-term impact of online behaviour. Cyber security concepts are introduced, including online threats, vulnerabilities and protecting personal data.	Students deepen their understanding of digital security and ethical use of technology. They explore cyber threats, security breaches and the ethical, legal and environmental impacts of computing systems. Students consider responsible use of computer systems and how legislation such as data protection influences computing practice.
Digital Literacy	Students develop confidence using a range of digital tools to create, manage and communicate information. They learn to evaluate the reliability and bias of online sources, interpret digital data and understand how online activity contributes to a digital footprint. Students are encouraged to use technology critically and responsibly when creating and sharing digital content.	Students apply digital literacy skills to interpret technical information, evaluate emerging technologies and communicate computational ideas effectively. They use appropriate technical vocabulary and apply their understanding to real-world contexts.
Creation and Programming	Students develop programming knowledge beginning with block-based coding before progressing to text-based languages such as Python. They apply programming concepts including sequencing, selection and iteration while creating increasingly complex digital products. Creative digital projects allow students to design and build functional artefacts such as animations, websites and interactive programs.	Students design, write and refine programs using text-based programming. They apply programming constructs including variables, selection, iteration, data types and subroutines to solve increasingly complex problems and develop robust, structured code.
Algorithmic Thinking	Students develop computational thinking by learning to break down problems into logical steps and designing clear instructions using flowcharts, pseudocode and programming tasks. They learn how algorithms underpin software, apps and everyday digital systems.	Students design, trace and evaluate algorithms using pseudocode and flowcharts. They explore standard algorithms such as searching and sorting and analyse algorithm efficiency when solving problems.
Problem Solving	Students use computational thinking to plan solutions, test programs and debug errors. They learn to approach problems logically and iteratively, refining solutions through testing and evaluation.	Students apply structured approaches to solving computational problems by identifying inputs, processes and outputs. They test, debug and refine programs while evaluating performance and accuracy.
Computing Fundamentals	Students develop understanding of key computing concepts including file management, networks, data representation and how digital systems store and transmit information. They also explore user interfaces, cyber security and the fundamentals of computer systems.	Students study computer system architecture including the CPU, memory, storage, operating systems, networks and data representation. They develop deeper understanding of how hardware and software interact and how computer systems function.

Assessment Objectives – Edexcel GCSE Computer Science

Assessment Objective	Descriptor
AO1 (30%)	Demonstrate knowledge and understanding of the key concepts and principles of computer science, including computer systems, networks, cyber security, data representation and programming concepts.
AO2 (40%)	Apply knowledge and understanding of computer science concepts to solve problems, including writing, tracing and interpreting programs and algorithms.
AO3 (30%)	Analyse problems in computational terms and evaluate solutions by designing algorithms, testing programs, identifying errors and refining solutions.

Vocabulary

Having a rich, ambitious, broad vocabulary is vital for students to succeed, both in school and throughout their lives. Tier 1 vocabulary is the simplest. These are the words we use in everyday conversation, such as “put”, “get”, “walk”, etc. Tier 2 vocabulary are challenging, ambitious words that don’t usually crop up in day-to-day conversation. These are the words that allow students to access academic texts, such as high-level literature, newspaper articles and exam papers.

Tier 3 vocabulary is the subject-specific vocabulary of a particular discipline. These are words that are uncommon outside of the context of a specific subject and enable students to communicate effectively within the subject. At Winifred Holtby Academy, tier 3 vocabulary is explicitly taught across our school curriculum and is mapped within the schemes of learning.

Assessment

Assessment in Computing is primarily formative at Key Stage 3 and embedded within lessons through practical programming tasks, digital projects, questioning and retrieval activities. Students demonstrate understanding through the creation of digital artefacts and problem-solving tasks. At Key Stage 4students complete exam-style questions and programming tasks to prepare for the Edexcel GCSE Computer Science examinations.

Careers and Real-World Applications (Computing)

The Computing curriculum provides students with opportunities to develop an understanding of how digital technologies shape the modern world and the skills required within the technology sector. Students explore how computer systems operate, how software is designed and developed, and how technology is used to solve real-world problems.

Through practical programming tasks, digital projects and problem-solving activities, students gain insight into a range of careers including software development, cyber security, data analysis, network engineering and digital design.

The curriculum supports students in developing transferable skills such as logical thinking, problem solving, computational thinking and collaboration. These skills are highly valued across a wide range of careers and support students in further study within computing, technology and related fields.

Curriculum Sequencing

Key Stage 3: Year 7 – Long Term Planning

	Autumn Term	Spring Term	Summer Term
Knowledge	**7.1 Introduction to Computing** • Logging into school systems and using Microsoft Teams • Managing files and folders effectively • Using keyboard shortcuts and saving work correctly • Email etiquette, formatting and attaching files • E-safety including responsible internet use and social media awareness • Baseline assessment using the skills developed **7.2 Spreadsheets** • Formatting spreadsheet data • Writing basic formulas and functions • Absolute and relative cell referencing • Creating charts to visualise data • Conditional formatting • Applying spreadsheet skills to solve problems	**7.3 Coding Skills – Scratch** • Introduction to programming environments • Sequencing instructions • Variables and operators • Selection and iteration • Decomposition and problem solving • Applying programming constructs through Scratch projects • Developing computational thinking through practical coding tasks	**7.4 Multimedia Development** • Creating presentations using PowerPoint • Using slide master, textboxes and shapes • Importing images and multimedia content • Developing online multimedia using tools such as Canva or Adobe Express • Creating and editing digital images using software such as GIMP or Inkscape • Combining media through hyperlinks, video and audio • Producing a final multimedia project

Key Stage 3: Year 8 – Long Term Planning

	Autumn Term	Spring Term	Summer Term
Knowledge	**8.1 Kodu Game Development** • Introduction to game design concepts • Using visual programming to control characters • Sequencing and event-driven programming • Designing game environments • Testing and refining gameplay **8.2 Flowol – Control Systems** • Understanding control technology • Flowcharts and system logic • Inputs, processes and outputs • Sensors and automated systems • Designing and testing control systems	**8.4 Theme Park Dashboard Project** • Designing dashboards using PowerPoint • Combining data and visuals • Presenting information clearly • Applying design principles • Evaluating digital presentations **8.5 Digital Media – Fake News Project** • Image editing using Photoshop • Understanding manipulation of digital media • Creating persuasive digital content • Evaluating reliability of online information	**8.6 Python Programming** • Variables and data types • Selection using conditional statements • Iteration using loops • Debugging programs **8.7 Creative Computing** • Animation using Pivot Animator • Robotics programming using VEXcode VR

	8.3 Spreadsheet Modelling • Advanced spreadsheet formulas and functions • Data modelling and analysis • Using logical functions • Data visualisation through charts • Evaluating spreadsheet models		

Key Stage 3: Year 9 students rotate through **all three units** every 13 weeks – Long Term Planning

	Year 9 Rotations (13 weeks each)
Knowledge	**9.1 User Interface Design** • Understanding user needs and audience • Principles of effective interface design • Layout, navigation and accessibility • Designing a quiz interface for Year 6 pupils • Evaluating usability and design choices **9.2 Animation Development** • Planning animations using storyboards • Creating animations using Pivot Animator • Timing, movement and sequencing • Refining and improving digital animations • Evaluating digital creative products **9.3 Website Development Project** • Designing website structures and layouts • Creating digital content using Adobe Express • Embedding media and examples • Communicating instructions and explanations clearly • Evaluating the effectiveness of digital content for an audience

Key Stage 4 Year 10 – Long Term Planning – EDEXCEL GCSE

	Autumn Term	Spring Term	Summer Term
Knowledge	**10.1 Computational Thinking and Programming Fundamentals** • Decomposition and abstraction • Algorithm design principles • Writing algorithms using pseudocode and flowcharts • Variables and data types • Selection and iteration • Trace tables and debugging **10.2 Python Programming** • Input and output operations • String manipulation • Conditional statements • Iteration using loops • Lists and arrays • Writing modular programs	**10.3 Computer Systems** • CPU architecture and components • Fetch–decode–execute cycle • Primary and secondary storage • Embedded systems • Operating systems and utility software **10.4 Data Representation** • Binary and hexadecimal • Binary addition • Binary shifts • Representing images and sound • Character encoding	**10.5 Computer Networks and Security** • Network types and topologies • Network hardware • Internet protocols • Cyber security threats • Prevention and detection methods **10.6 Programming Development** • File handling • Advanced iteration and nested structures • Testing and debugging strategies • Developing structured programs

Key Stage 4: Year 11 – Long Term Planning – EDEXCEL GCSE

	Autumn Term	Spring Term	Summer Term
Knowledge	**11.1 Algorithms** • Searching algorithms (linear and binary) • Sorting algorithms (bubble, merge, insertion) • Algorithm efficiency • Tracing and evaluating algorithms **11.2 Ethical, Legal and Environmental Impacts** • Data protection and privacy • Computer misuse legislation • Ethical implications of technology • Environmental impacts of computing	**11.3 Advanced Programming** • Modular programming • Subroutines and functions • Validation and error handling • Working with complex data structures • Debugging and testing strategies **11.4 Exam Preparation** • Applying computational thinking to exam problems • Writing exam-style algorithms • Trace tables and program analysis	**11.5 Revision and Consolidation** • Revisiting core programming constructs • Review of computer systems and networks • Practice exam questions • Retrieval practice and knowledge consolidation • Exam technique and timed responses