Computing Department KS3 Subject Information

For more information, please consult: M. Gómez

HOD name, job title and email M. Gómez, Head of Computing, mgomez@gilesacademy.co.uk

Learning Aims / Learning Objectives

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

DfE. Statutory guidance. National curriculum in England: computing programmes of study. Published 11 September 2013

What will I be learning and progressing to each year?

Year	Key Knowledge	Key Skills	Key Vocabulary
7	**Unit 1** - Understand that the verb 'to compute' can be applied to any mathematical calculation. Understand that there are many ways to compute and that the modern electronic computer was developed to carry out computations at enormous speed. Be able to name the main components of a computer. Understand that all computation requires some form of input (e.g. a number), a process (e.g. a calculation) and produces an output (the answer). Know that there are different number systems, decimal, binary, etc. Know that data is stored in computers in units called bytes. Know that a byte is made up of eight smaller units called bits. Understand that data must be converted to digital format to be processed by a computer.	Logical reasoning to predict the behaviour of simple programs. Simple arithmetic.	Data, Compute, Input device, Memory, Storage device, Processor, Output device, Central processing unit (CPU), Decimal, Binary, Bit, Byte.
	Unit 2 - Understand that problems are easier to solve if broken down into smaller parts. Understand that complicated activities can be recorded as a sequence of simple instructions. Know that a sequence of discrete instructions written to solve a problem is called an algorithm. Know that algorithms must be clear and unambiguous. Be able to write a simple algorithm. Be able to correctly apply the terms 'decompose', 'data collection' and 'algorithm'. Be able to identify the data relevant to the solution of a problem. Be able to decompose a problem into smaller problems. Be able to collect and organise data to solve a problem. Be able to use a spreadsheet template with built in formulas for data modelling and analysis. Be able to design simple algorithms.	Computational thinking skills: abstraction, decomposition, pattern recognition and algorithmic thinking to solve a problem. Problem solving. Logical reasoning.	Decompose, Algorithm, Computational thinking, Data Information, Pattern identification, Hypothesis testing, Model, Abstraction, Generalisation

	Unit 3 - Know and understand the basic features of regular shapes including sides and angles and their relationships. Understand how patterns define relationships between objects and the concept of repeating patterns. Know and understand how to write algorithms to create basic geometrical shapes. Be able to understand and explain the key concepts of decomposition and abstraction. Know and understand how to draw basic geometrical shapes using a graphical programming language. Understand how written algorithms can be translated into a graphical programming language.	Geometry. Computational thinking skills: abstraction, decomposition, pattern recognition and algorithmic thinking to solve a problem. Problem solving. Logical reasoning.	Abstraction, Generalisation, Geometrical shapes, Decomposition, Algorithm, Coordinates, Iteration, Graphical programming
	Unit 4 - Be able to identify some of the steps in an algorithm for specific task. Be able to understand the need for accuracy when issuing instructions. Know the types of task best performed by humans and the types of task best performed by computers. Be able to create part of a dance routine using an iteration loop. Know that a procedure is a sequence of instructions that can be called on and executed when required. Know what pattern recognition is and use it when evaluating a dance routine. Be able to create blocks of dance sequences using iteration loops. Know that selection provides possible courses of action that will result when certain conditions are met. Be able to input values and use these to control a dance sequence.	Creativity. Computational thinking skills: abstraction, decomposition, pattern recognition and algorithmic thinking to solve a problem. Problem solving. Logical reasoning.	Algorithm, Dry run, Execute, Sequence, Iteration, Procedure, Function, Procedural abstraction, Selection
	Unit 6 - Appreciate that everything on the web is interlinked. Understand that the web is like a library. Understand how web pages are related. Understand the difference between the WWW and the internet. Be able to identify a poor quality web page and avoid using or relying on it Understand what is meant by the term 'bogus'. Understand the difference between fact and opinion. Appreciate when a web site could be dangerous. Understand that searching is the main way to use the web. Be able to obtain purposefully information on a topic on the web. Appreciate that search engines collect information about the web and use algorithms to rank websites in search results.	Critical thinking. Digital skills.	Internet Server, Client, Web browser, Artefact, Network, World wide web (WWW, the web), HyperText Markup Language (HTML), Protocol, Search engine, Boolean operators
	Link to knowledge organiser here	**Link to homework here**	**Link to full Y9 vocabulary here**

gilesacademy

	The content of your curriculum in this academic year for your subject **Unit 1** - Outline of the history of computing and the electronic computer. Practical study of computer hardware to introduce students to the components that make up a computer. Functions of the component parts. Function machines at basic level.E.g. take a number (such as 4) as INPUT, apply a process (such as 'multiply by 3') and give another number as OUTPUT (in this case, 12). The role of Bletchley Park in shortening the Second World War and the part played by Colossus in breaking codes. Electrical circuits. Morse code as a method of using electrical signals to send messages and using codes to represent characters. Morse code uses long and short signals and spaces to represent characters, whilst digital computing uses simple, yet elegant, on/off binary patterns. Similarities and differences between the two systems and why the more complex Morse code system is not a suitable code for processing data electronically (because, in addition to the two states of 'on' and 'off', Morse code contains a third factor that is required to distinguish between a dot and a dash – time). Hardware and processing. Data and data representation. **Unit 2** - Understand that computational thinking is not 'thinking like a computer' but thinking about problems (and the world) in terms of the processes going on, the data available, and the steps that need to be followed in order to achieve a goal. They should understand that while human beings can use context and common sense to interpret instructions, can ask for clarification, and can act on their own initiative, computers cannot do any of these things. They should also understand that humans are best employed doing creative and innovative things, while computers are best used for repetitive tasks that require speed and precision. Application of a simple version of computational thinking that can be broken down into a small number of discrete activities. Decomposition and algorithms skills involved in solving problems by thinking logically. They will be revisited in a number of subsequent units, when the concepts touched on here will be explained in much greater depth. The emphasis at this stage is on having fun and the activities have been designed to make the subject as playful and engaging as possible. **Unit 3** - You will be getting students to appreciate the link between maths, art and computer science, particularly the creativity and imagination required to create works of art and computer programs based on both artistic and mathematical concepts. Students will understand simple algorithm design and the importance of being able to identify the important ideas (abstraction) and breaking down the problem into manageable units (decomposition). Students will also be introduced to repetition (iteration) as one of the key constructs in programming. You will help them discover how to design algorithms for some basic shapes. You will be getting students to write programs to draw shapes they have defined using algorithms. It is important not to confuse algorithms and code. An algorithm is a set of rules that defines a solution. It is possible to write an algorithm without using iteration and implement it in code. However, for the purposes of this unit, writing the algorithm using iteration should follow through into the code. This is followed up in Lesson 3 with a practical session that requires students to code their artworks. **Unit 4 –** Students will be using a graphical programming language to recreate a dance sequence. They are asked to take and use images of themselves in various positions to recreate these moves. You will need to show them how to upload these images for use in your chosen graphical programming language. You will also need to be able to demonstrate the basic constructs within the programming language, including wait commands and 'forever if' loops. Note that some students might not wish to use photographs of themselves, and suitable images could be provided for them to use. Students program their dance animation using more than one dance sequence, and need to become familiar with repeating, selecting, iterating and procedures in your chosen graphical programming language. **Unit 6** - The power and complexity of interlinked content on the web and how navigation works. Folder– subfolder structure of websites and a basic HTML tags. Content of websites and issues raised about e-safety. Relative accuracy and reliability of websites. How weblinks work. Boolean operators (AND, OR and NOT), and how Venn diagrams can be used to help understand how they work.

Year	Key Knowledge	Key Skills	Key Vocabulary
8	**Unit 1** Be able to describe what is meant by an operating system. Know the names of a range of operating systems. Be able to describe examples of different operating systems.	Writing skills,	Operating system, BIOS, Driver, Translator
	Unit 2 (partially) Be able to create, move and rename files and folders Be aware of at least one method of copying and pasting files. Understand that sensible file and folder names make it easier to find work later.	Logical thinking, digital skills	
	Unit 3 Be able to convert a decimal number to binary (1–255) Be able to convert a binary number to decimal (1–255) Be able to count in binary to 8 bits. Be able to add two 8-bit binary numbers together Understand that binary numbers are held in 8-bit fixed bit strings on computers. Be able to explain how to convert binary to decimal and vice versa. Produce a tutorial on an aspect of binary arithmetic Evaluate and give feedback on another person's work using a guide.		Integer, Binary, Decimal, Bit, Byte, Overflow
	Unit 5 Understand that problems need to be broken down into more manageable parts. Understand how selection statements can be used to provide alternate paths in a solution. Be able to sequence a set of selection statements. Be able to combine selection and the Boolean operators AND and OR. Be able to use variables as a method of recording incremental changes. Be able to nest 'if' statements to enable two inputs to be checked. Understand that Venn diagrams can be used to demonstrate graphically nested selection sequences. Be able to use the NOT operator. Be able to use 'if' statements with AND. Understand that Boolean logic combines Boolean operators. Be able to identify the outputs of a truth table for a group of connected Boolean operators. Be able to use 'if else'. Understand how different forms of 'if' statement can affect the organisation of programs. Be able to identify where loops can be used to repeat commands. Understand that there are a variety of different loops that are used in different situations. Understand that there are a range of standard algorithms that have been designed for mass use. Know that two algorithms that solve the same problem may not be the same. Be able to adjust algorithms to deal with exceptions	Arithmetic, computational thinking, logic, creativity	Selection, Decompose, Sensor, Abstraction, Boolean, operators, Nesting, Venn diagram, NOT Boolean logic, if else, Nesting, Venn, diagram, NOT Loop, Algorithm, Programming, Path finding, Procedure

	Unit 6 Know and understand the common ways to connect to the internet. Understand the role of the ISP. Understand the relationship between LANs and WANs. Understand how data streams generated by computer applications are split into packets. Know and understand the role of routers. Be able to distinguish between circuit switching and packet switching. Know the four TCP/IP layers. Understand that the four TCP/IP layers are independent of each other. Understand that data can become corrupted.	Analytical skills, digital skills,	LAN WAN ISP Metadata Circuit switching Packet switching IP address Domain Name System (DNS) None Flag
	Unit 7 Be able to explain what is meant by a sorting algorithm. Be able to export data to and from a list/array in Scratch. Be able to describe how the Bubble Sort algorithm works. Be able to code the Bubble Sort algorithm with structured guidance. Be able to use iteration to move through a list one item at a time. Be able to swap the contents of two variables via an intermediary variable. Be able to code the Bubble Sort algorithm with structured guidance. Be able to describe how the Selection Sort algorithm works. Be able to code the Selection Sort algorithm with structured guidance. Write code to find the smallest item in a list. Understand that 'nesting' is the process of placing a section of code within another section of code.	Algebra, Computational Thinking: algorithmic thinking, decomposition; logical reasoning	Ascending order Descending order Pseudocode Nesting
	Link to knowledge organiser here	**Link to homework here**	**Link to full Y10 vocabulary here**

The content of your curriculum in this academic year for your subject

Unit 1 What is an operating system? (Looking at layers of abstraction); Different types of operating system; and Grouping operating systems. Main features of an operating system, common similarities and differences between operating systems. Difference between proprietary and open source operating systems.

Unit 2 (Partially CMD line as well as file management commands in the CMD line. Older people taking their first steps on the internet, and good methods of file management. Students must be able to create, move and rename files and folders; be aware of at least one method of copying and pasting files; and understand that sensible file and folder names make it easier to find work later. Be aware of file extensions and their purpose. To familiarise themselves with the command line prompt, and understand that there are two ways to manage files and folders. To identify advantages and disadvantages of both graphical and command line methods for managing files and folders. Suggest situations where the command line would be preferable to a GUI for file manipulation.

	Unit 3 Understand and use binary digits, such as to be able to convert between decimal and binary. Students will know the value of each bit of an 8-bit binary string. To be able to convert a decimal number to binary (1-255); and to convert a binary number to decimal (1-255); and be able to count in binary to 8 bits. To perform simple binary addition with two 4-bit binary number. Create, reuse, revise and repurpose digital artefacts for a given audience, with attention to trustworthiness, design and usability. Students will produce a tutorial on an aspect of binary arithmetic; and evaluate and give feedback on another person's work using a guide. Most students will be able to produce a video tutorial on an aspect of binary arithmetic and make adjustments to the work based on the feedback received. **Unit 5** Design, use and evaluate computational abstractions that model the state and behaviour of real- world problems and physical systems. They will be using selection and decomposition. 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. Make appropriate use of data structures such as lists, tables or arrays; design and develop modular programs that use procedures or functions. To be able to combine selection and the Boolean operators AND and OR; and be able to use variables as a method of recording incremental changes. Students will be able to nest 'if' statements to enable two inputs to be checked; understand that Venn diagrams can be used to demonstrate graphically nested selection sequences; and be able to use the NOT operator. Use 'if' statements with AND; understand that Boolean logic combines Boolean operators; and be able to identify the outputs of a truth table for a group of connected Boolean operators. To use 'if else'; and will understand how different forms of 'if' statement can affect the organisation of programs. Explore loops and patterns, and where loops can be used to repeat commands; and that there are a variety of different loops that are used in different situations. To be able to create a procedure to deal with an exception, and understand how a procedure can be represented in a flowchart. **Unit 6** 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. Understand the hardware and software components that make up computer systems and how they communicate with one another and with other systems. Understand how people connect to the Internet, past, present and future; and understand the role of the ISP. To understand the relationship between LANs and WANs. To understand packets of data, and IP addresses. How data streams generated by computer applications are split into packets; understand the role of routers; and be able to distinguish between circuit switching and packet switching. Know the four TCP/IP layers; understand that the four TCP/IP layers are independent of each other; and understand that data can become corrupted. **Unit 7** Understand several key algorithms that reflect computational thinking such as ones for sorting and searching; 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 such as lists, tables or arrays; design and develop modular programs that use procedures or functions. Understand sorting in ascending and descending order, and BogoSort and Swap Sort. Be able to explain what is meant by a sorting algorithm and how the size of a data set affects the time it takes for an algorithm to sort data. Understanding Bubble Sort and its limitations. To use iteration to move through a list one item at a time; and swap the contents of two variables via an intermediary variable. To know that pseudocode is used in program design and is a system of writing that resembles a simplified programming language.
How will my work be assessed? / assessment components / frequency / term	
Y7	Baseline assessments will be done before each topic. You will be tested again after the topic has been taught. One unit per term.
Y8	Baseline assessments will be done before each topic. You will be tested again after the topic has been taught. One unit per term.

Extra-curriculum activities / Trips / Community cohesion / Events participation
Homework/revision club every Monday in IT1. Coding Club every Thursday first lunch in IT1.
What qualifications and career paths this subject will enable me to access in KS4? KS4 option subjects / Career Paths
GCSE Computer Science can lead to A level qualifications and university before you decide to follow your career path as: Software developer**,** Database Administrator, Computer Hardware Engineer, Computer Systems Analyst, Computer Network Arquitect, Web Developer, Information Security Analyst, Computer programmer, Project Manager **Tech Award in Digital Information Technology can lead to Level 3 qualifications, apprenticeships and university before you decide to follow your career path as:** Technical Support, Computer programmer, Web Developer, Computer Systems Analyst, Data Analyst, IT Security, Network Engineer
How parents or other members of the public can find out more about the curriculum your subject is following
The Computing Department follows the Key Stage 3 national curriculum in England. https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study/national-curriculum-in-england-computing-programmes-of-study Dynamic Learning. *(students have their own username and password to access the online curriculum)* https://my.dynamic-learning.co.uk/Default.aspx?ReturnUrl=%2fMyDynamicLearning.aspx
Image