



Curriculum Information 2017-18

COMPUTER SCIENCE

1. Key Stage 3

Computer Science KS3 Topics						[Staff contact: Ms Akpojaro]
	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
Year 7	<p>7.1 Introduction to Computing- File management, Hand-in own cloud, Baseline test, use of different types of software,</p>	<p>7.1 Introduction to Computing- Multi[plication Binary, code binary convertor using Scratch</p>	<p>7.3 ICT (IT applications) Problem solving with spreadsheets: Students will be analysing data and meeting the needs of known users</p>	<p>Digital Literacy 7.2 – Issues of computer use - In this unit of work, students will learn how to use the internet safely and effectively. They will learn about copyright law, search engines (including the use of Boolean logic for effective searching) and they will also learn about the dangers of the internet and ways to combat these dangers.</p>	<p>7.4 Introduction to HTML-This unit teaches the basics of HTML enabling students to create a mini website. Students learn how to add text, images and hyperlinks, plus formatting techniques including fonts, text size and alignment</p>	<p>7.5 Introduction to python programming using MicroBit -This unit introduces students to the MicroBit The unit uses 'Python' programming language.</p>
Year 8	<p>8.1 Computer hardware - The computer hardware unit is designed to teach students what a computer system is, the various components of a computer system and their purpose. Students will also learn about the purpose of the CPU, RAM, Hard Drive and I/O devices and how the all function together and the function of the CPU, including the fetch, decode, execute cycle.</p>	<p>8.2 Binary Bits and Bobs Binary Bits and Bobs introduce students to the binary number system, converting between binary and denary and simple binary addition. Students will also be taught how (and why) characters, images and sound are represented by the binary system</p>	<p>8.3 Algorithms - Computational Thinking: -abstraction -decomposition -algorithmic thinking How to produce Algorithms using: -pseudocode -using flow diagrams</p>	<p>8.4 HTML and CSS - ADU Students will be reminded of some basic HTML syntax (as covered in the year 7 unit) and will be introduced to CSS so that they can understand how to better present their webpages. They will learn how to add gradient backgrounds, add page borders, curve images and reorganise content on the page with the help of DIV tags.</p>	<p>8.5 Introduction Python - In this unit, students will be introduced to programming in the Python programming language. They will learn how to print messages to the screen, ask the user to input data and stores this data in variables. They will also understand how computers make decisions and consequently learn how to program IF statements</p>	<p>8.6 The Digital project- For this project students will work as part of a group to plan and create a digital product.</p>
Year	<p>More python programming-</p>	<p>Intermediate python programming-</p>	<p>9.2 Hardware and Networks-Students will be</p>	<p>9.3 Binary and Data Representation</p>	<p>9.4 Algorithms</p>	<p>9.5 Advance Python programming with GUI</p>



Curriculum Information 2017-18

9	<p><i>Continuing on from the year 8 unit of work which introduced the Python programming language, students will reinforce their understanding of inputs, outputs, variables and selection through the means of a variety of programming challenges. Students will also be taught the programming structure of iteration. They will learn how FOR and WHILE loops work and will code these structures in a range of programs.</i></p>		<p><i>introduced to Local Area Networks (LANs), the hardware of a local network, the workings of the Internet, how the WWW and Internet differ and how data travels around a network (e.g. Data Packets).</i></p>			<p><i>Building with TKinter</i></p>
----------	---	--	---	--	--	--

Computer Science Assessment in Key Stage 3:

In Computer Science, you will receive detailed written feedback on the following pieces of work this year. There will be opportunities for you to respond to that feedback.

Term	Year 7	Year 8	Year 9
Autumn	Introduction to Computing: Scratch Binary Converter coding	Hardware: Hardware end of unit Test	Python: Coding end of unit test
Spring	Issues of computer use: Written assessment	Algorithms - Computational Thinking: Written Assessment	Hardware and Network: Written Assessment
Summer	End of Year Exams: 15-29.06.18	End of Year Exams: 15-29.06.18	End of Year Exams: 11-22.06.18



Curriculum Information 2017-18

Keywords and Subject Specific Vocabulary					Computer Science	
	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
Year 7	Logon, Password, Save, Folder, Bit, byte, binary, denary, ASCII; Hexadecimal	Sequence inputs, Motion, Operator, Variable, Start, Flag, Looks, Sprite, Costumes Pen, Control, Forever, Repeat, Script	Cell, column, Sum, Average, Worksheet, Formulas, Referencing, Absolute, Formatting, Conditional formatting, What if, Validations	Evaluate, Trustworthiness, Bias, Reliability, Copyright, Acknowledgement, Plagiarism Social media, privacy setting, digital footprint, E-safety	HTML, Tags, Browser, Hyperlink, Navigation, Table	Microbit, , variables Compile, Flash, coordinates, syntax Accelerometer Python shell
Year 8	Input device, output device, Storage, CPU, RAM, Motherboard	Binary , Denary, Hexadecimal, Bit, Byte Nibble	Algorithm, Sequence, Planning, Flowcharts, Loops, Decision, Process, Input, Output	HTML, Tags, CSS, DIV Tags, Internal , External, Style sheet	Data types, Float, Int Define, Variable, Function Print, IF statements Conditionals	Research, Design, Digital, Design Tools, Evaluation, Justify, audience
Year 9	Input, Variable, Output, Syntax, Errors, Decision, Integer, Iteration, While loop	String, Constant, Validations, Statement, Comment, Nested if	Networks, LAN, Network, Inter-face card, Switch, internet, WAN, Data Packets	Addition, Carry, Compression, Sampling	Pseudocode, flowcharts, start/end, execute,	Function, Procedures Def, Import module

How to support your son at home			Computer Science
What sorts of independent work/homework will he get?	How much help should you give him?	What are the top three tips for supporting independent learning?	Useful resources and links
<p>Homework can be to practise skills currently being developed in class.</p> <p>It may be to find, watch, and make notes on tutorial videos.</p> <p>It may be testing skills, knowledge, and understanding developed in class.</p>	<p>It is good for students to complete homework independently, but it is very useful to ask him to explain the homework to you, or to ask questions about it.</p> <p>If the homework covers a topic that you know well, asking him leading questions is a good way to help him to work out the answers.</p>	<ol style="list-style-type: none"> 1. Discuss the lessons and the homework, asking him to teach you what he has been learning. 2. Encourage him to use the support resources provided, and the internet as a whole to research the topic. 3. Many of the programs that we use are available either through the school website (My Programs) or can be downloaded and installed at home for free. Having these available at home means that he can develop his skills at any time. 	<p>The computing department website: http://computing.foresthillschool.co.uk This site has a huge range of lesson resources, and contains links to further websites that contain even more support resources.</p> <p>http://www.codecademy.com This site has a number of free online courses in a range of languages, including html, css, and python.</p> <p>scratch.mit.edu This site is an online Scratch editor, and allows programmers to share code and help each other out with their programming.</p>



Curriculum Information 2017-18

2. Key Stage 4

GCSE Computing Topics					[Ms Akpojaro & Mr Simpson]	
	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6
Year 10	Unit 2:1 Algorithms Unit 2:2 Programming Techniques	Unit 2.3 Producing Robust Programs Programing	Unit 2:5 Translators and Facilities Sample Program Assessment	Unit 2.6 Data Representation Programming Development	Practice NEA Assessment	Unit 1- Computer systems, Operating systems, utility software systems
Year 11	Unit 1- Computer systems	Practice Controlled Assessment Task	Unit 1 How to investigate and discuss computer science technologies	Unit 1 Legislation relevant to computer science	Revision and Exam Practice	

GCSE COMPUTING Assessment:

Term	Year 10	Year 11
Autumn	Coding Challenge – Written Feedback	Pre-public exams: 04-15.12.17
Spring	Computational thinking, algorithms and programming	Jan 8 th to Feb 23 rd Controlled Assessment – 20% of final Grade
Summer	Pre-public exams: 09-20.07.18	Public exam dates:

KS4 Topics: ICT					[Mr Simpson/ Ms Akpojaro]	
	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6
Year 10	ADU: Purpose and properties of digital graphics (R082) PSI: Purpose and content of pre-production (R081)	ADU: Plan the creation of a digital graphic (R082) PSI: Purpose and content of pre-production (R081)	ADU: Create a digital graphic (R082) PSI: Planning pre-production (R081)	ADU: Review a digital graphic (R082) PSI: Producing pre-production documents (R081)	ADU: Uses and properties of interactive multimedia products (R087) PSI: Reviewing pre-production documents (R081)	ADU: Plan interactive multimedia products (R087) PSI: Properties and features of multipage websites (R085)
Y11	VAO -RO04	VAO-RO04	VAO-R001	VAO-R001	External exam	



Curriculum Information 2017-18

ICT Assessment:

Term	Year 10	Year 11
Autumn	1: Ongoing coursework for R082 (LO1-2) 2: Tailored exam paper using unit specific questions	Coursework Catch-up/improvement
Spring	1: Ongoing coursework for R082 (LO3-4) 2: Tailored exam paper using unit specific questions	1: On-going mini test in preparation for external exam 2: Mock test
Summer	Pre-public exams: 09-20.07.18	Public exam dates:

KS4 How to support your son at home

Computer Science

What sorts of independent work / homework will he get?	How you can help	Useful resources and links
Homework will largely be in the form of questions based on theory topics covered in the lessons, research planning and practical activities.	<ul style="list-style-type: none"> Provide access to Python IDLE Encourage your son to program for fun using online Python tutorials Discuss and create a revision schedule with students Act as a tester to check students learning Purchase for and encourage your son to use the revision guide: 'My revision notes – OCR Computing for GCSE', Hodder Education, O'Bryne and Rouse, 2013. ISBN 978 1 444 193848 	Exam board course link: Recommended revision guide: http://www.ocr.org.uk/qualifications/by-subject/computing/computing-resources/ Revision materials in Hand-in and Computer Science\GCSE Computing Year 11\Revision <ul style="list-style-type: none"> GCSE bbcbite Size: http://www.bbc.co.uk/education/subjects/z34k7ty

Curriculum Information 2017-18



KS4 How to support your son at home

ICT

What sorts of independent work / homework will he get?	How you can help	Useful resources and links
Homework will largely be in the form of improvements by responding to teacher feedback on coursework.	<ul style="list-style-type: none">• Check your son has completed R002/R003 and R004.• Check that your son attend catch up session for coursework• Discuss and propose a revision schedule for the exam unit	<p>http://www.ocr.org.uk/qualifications/creative-imedia-level-1-2-award-certificate-j807-j817/</p> <p>http://ocr.org.uk/qualifications/cambridge-nationals-ict-level-1-2-j800-j810-j820/</p>



Curriculum Information 2017-18

3. Assessment Criteria (KS3 and 4)

Year 7 Steps Assessment Criteria					
	Introduction to Computing	Issues of Computer Use	Spreadsheets	Introduction to HTML	Python Programming with MicroBit
Step 4	<ul style="list-style-type: none"> I can recommend appropriate assistive input and output devices based on user need I can add binary numbers 	<ul style="list-style-type: none"> I can explain why some data is unsafe online and what to do instead I can explain the possible outcomes of cyberbullying for both perpetrator and victim 	<ul style="list-style-type: none"> I can organize data to make it usable in calculations I can use complex functions like VLOOKUP or IF 	<ul style="list-style-type: none"> I can explain DIV Tags I can create a webpage which has been formatted using CSS. I can create a webpage which contains divisions and where CSS code applies different styles to the different divisions. 	<ul style="list-style-type: none"> I can use the if-else statement accurately in my programming I can use a while loop in a program I can correct run-time errors that are highlighted by the computer when the program is run
Step 3	<ul style="list-style-type: none"> I can explain the difference between input and output devices I can convert numbers between binary and denary without a table of column values I can explain the difference between hardware and software 	<ul style="list-style-type: none"> I can identify a range of unsafe data online I can carry out research and write up the results in my own words I can explain the difference between masquerading and anonymity 	<ul style="list-style-type: none"> I can create a graph or chart with a title and labels I can use simple functions like SUM or ROUND 	<ul style="list-style-type: none"> I can control the font size and text colour of elements in a webpage. I can use anchor tags within my webpage. 	<ul style="list-style-type: none"> I can use input and print commands in a program using suitable variable names I can combine text, numbers and variables in a single output line I can use the if – else statements in a program
Step 2	<ul style="list-style-type: none"> I can categories input, output and storage devices I can convert numbers between binary and denary with the help of a table of column values I know the difference between hardware and software 	<ul style="list-style-type: none"> I know multiple ways to stay safe online I know how to credit research sources I can give an example of unsafe data online I can give examples of cyber-bullying 	<ul style="list-style-type: none"> I can resize cells and adjust their borders I can carry out mathematical calculations I can create a simple graph or chart 	<ul style="list-style-type: none"> I can explain HTML I can make images appear in a webpage. I can use tags to emphasise certain words or phrases in a webpage. I can add links to other webpages in my webpage. 	<ul style="list-style-type: none"> I can give simple, accurate instructions for another person to follow I can explain why the int() and str() commands need to be used in programming I can correct simple syntax errors such as a missing “ or)
Step 1	<ul style="list-style-type: none"> I can name an input device I can name an output device I can name a storage device I can organize my files and folders 	<ul style="list-style-type: none"> I know what plagiarism is I know what copyright means I know what cyber-bullying is I know what to do if I feel unsafe online 	<ul style="list-style-type: none"> I know the difference between data and labels I can enter data accurately into a spreadsheet I understand the terms 'cell', 'row', and 'column' 	<ul style="list-style-type: none"> I can make text appear in a webpage I can use heading and paragraph tags 	<ul style="list-style-type: none"> I can write a python program that displays text on a MicroBit I can write a python program that displays a sequence of numbers on a microbit



Curriculum Information 2017-18

Year 9 to 11 Assessment Criteria				
Step	9.1 and 9.5 Python Programming	9.4 Algorithms	9.2 Hardware and Networks	9.3 Binary and Data representation
Step 9	<ul style="list-style-type: none"> I can create procedures that call procedures, to multiple levels. (Building one-abstraction on top of another) I can create programs that read and write persistent data to files 	<ul style="list-style-type: none"> I understand searching algorithms – binary search, linear search I understand sorting algorithms – bubble sort, merge sort and insertion sort 		
Step 8	<ul style="list-style-type: none"> I can document programs to help explain how they work 	<ul style="list-style-type: none"> I know how to interpret, validate, test, correct or complete algorithms 	<ul style="list-style-type: none"> I understand the components that make up digital systems, how they communicate with one another and with other systems I can compare wired and wireless networks I can explain network topologies and protocols 	
Step 7	<ul style="list-style-type: none"> I can create, test and evaluate programs against user requirements 	<ul style="list-style-type: none"> I can use logical reasoning to compare the utility of alternative algorithms for the same problem 	<ul style="list-style-type: none"> I understand how computer networks can provide multiple services, e.g.: email, instant messaging 	<ul style="list-style-type: none"> I understand how numbers can be represented in binary and be able to carry out simple operation on binary numbers, e.g. binary addition, conversion between binary and decimal
Step 6	<ul style="list-style-type: none"> I can design and develop modular programs that use procedures or functions I can solve problems by decomposing them into smaller parts in a language I can make appropriate use of data structures e.g. arrays 	<ul style="list-style-type: none"> I understand that algorithms may be decomposed into components parts(procedures), each of which itself contains an algorithm I can use logical reasoning to detect and correct errors in algorithms I can design, write and debug programs that accomplish specific goals 	<ul style="list-style-type: none"> I can demonstrate an understanding of what the internet is and what data packets are I can explain how data travels around the internet I am able to explain the various parts of a data packet 	<ul style="list-style-type: none"> I understand how text, images and sound can be represented digitally in the form of binary numbers, e.g. 2 bit image
Step 5	<ul style="list-style-type: none"> I can use for and while loops in my programs to repeat commands I can correctly create a program from a flowchart I can correct logic errors where a program looks like it works but is producing the incorrect answer 	<ul style="list-style-type: none"> I understand that algorithms are implemented as programs I can write algorithms with care and precision to avoid errors and ambiguity 	<ul style="list-style-type: none"> I can demonstrate an understanding of what a network is I can explain the advantages and disadvantages of networking and will state some devices needed for networking I am able to explain the devices needed to create a local area network. 	<ul style="list-style-type: none"> I can add up three binary numbers I can explain how bitmaps are stored using binary and why images can get pixelated



Curriculum Information 2017-18

				<ul style="list-style-type: none"> • I can explain the difference between bitmap images and vector images • I Understand how sound is represented in a computer • I can complete a simple truth table based on a single logic gate
Step 4	<ul style="list-style-type: none"> • I can use the if-else statement accurately in my programming • I can use a while loop in a program • I can correct run-time errors that are highlighted by the computer when the program is run 	<ul style="list-style-type: none"> • I can use sequence, selection and repetition in flowcharts/pseudocode • I can solve problems by decomposing them into smaller parts • I can correct algorithms if they fail tests 	<ul style="list-style-type: none"> • I understand computer networks including internet • I understand how computers communicate with other systems 	<ul style="list-style-type: none"> • I can add up two binary number and then convert the answer to denary to check calculation • I can convert between binary and hexadecimal • I can convert binary into text using an ASCII conversion table • I can convert binary numbers to images
Step 3	<ul style="list-style-type: none"> • I can use input and print commands in a program using suitable variable names • I can combine text, numbers and variables in a single output line • I can use the if – else statements in a program 	<ul style="list-style-type: none"> • I understand that algorithms are implemented as programs • I can write algorithms with care and precision to avoid errors and ambiguity 	<ul style="list-style-type: none"> • I know the hardware and software components that make up computer systems 	<ul style="list-style-type: none"> • I can convert binary numbers to their denary equivalent without a table to help me place the value • I can add up two binary numbers by converting the values to denary first • I can convert hexadecimal numbers to denary and back again • I understand how an image is represented in a computer