Alcester Academy Curriculum Planning: Computer Science GCSE Year 11

Department: *ICT & Computing (Computer Science)*						
Term	Topic/Subject	Assessment Objectives and Knowledge	Skills	Literacy, Numeracy (including wider reading)	Personal Development (SMSC, British Values, Careers, Healthy Living, Citizenship Equality and Diversity, Preparation for next stages	AFL/Summative Assessment
Autumn 1	NEA Programming Task	The Programming Project requires learners to use skills from Component 01 and Component 02 to create a solution to a set problem. They will code their solution in a suitable programming language. The solution must be tested to ensure they solve the stated problem. Learners must create a suitable test plan with appropriate test data.	Learners demonstrate their practical ability in the skills outlined in the specification, supporting the learning of Components 01 and 02.			The Programming Project task will require candidates to extract the key elements from the scenario set and to create logical designs – either explicitly through pseudocode or flow charts, or through design when creating code directly. Their ability to do this will reflect how confidently they will be able to read an algorithm question within the examination paper and formulate a suitable response.
Autumn 2		• how to identify and use variables, operators, inputs, outputs and assignments • how to understand and use the three basic programming constructs used to control the flow of a program: Sequence; Selection; Iteration • how to understand and use suitable loops including count and condition controlled loops • how to use different types of data, including Boolean, string, integer and real, appropriately in solutions to problems • how to understand and use basic string manipulation how to	Thinking abstractly – removing unnecessary detail from the problem, and Control and Data abstraction Thinking ahead – identifying preconditions and inputs and outputs Thinking procedurally –			

		understand and use basic file handling operations: open, read, write, close • how to define and use arrays (or equivalent) as appropriate when solving problems • how to understand and use functions/sub programs to create structured code.	identifying components of problems and solutions Thinking logically – predicting and analysing problems Thinking concurrently – spotting and using similarities.			
Spring 1	2.6Data Representation	• Understand what is meant by the terms bit, nibble, byte, kilobyte, megabyte, gigabyte, terabyte and petabyte. • Know how to represent the capacity of data storage using these units, and be able to convert between them. • Understand that data needs to be converted into a binary format to be processed by a computer. • Know how to perform a left and right binary shift. • Understand what binary shift achieves. • Know how to convert positive denary whole numbers (0–255) into 2 digit hexadecimal numbers and vice versa. • Know how to convert from binary to hexadecimal equivalents and vice versa. • Know what a check digit is. • Understand why check digits are used. • Understand that all data must be represented in binary numbers, including text. • Know what is meant by the term “character set”. • Understand the relationship between the number of bits in the character set and the number of characters that can be represented. • Understand how an image is represented as a series	• Know how to convert positive denary whole numbers (0–255) into 8 bit binary numbers and vice versa. • Know how to add two 8 bit binary integers. • Understand how overflow errors occur.	Bit, Nibble, Byte, Kilobyte, Megabyte, Gigabyte, Terabyte, Petabyte Denary, Binary, Hexadecimal, Binary addition overflow, Binary shift, Check digit, Character set, ASCII, Unicode, Pixel, Metadata, Colour depth, Resolution, Bit rate, Sampling frequency, Compression, Lossy compression, Lossless compression		Assessment & Feedback through topic based questions, and through practical programming activities. Feedback provided on pupil workbooks. End of topic tests for each topic through exa foundation/craigndave resources & smart revise software.

		of pixels represented in binary. • Know what is meant by the term 'metadata' and be able to give examples. • Understand the effect of colour depth and resolution on the size of an image file. • Understand how an image is represented as a series of pixels represented in binary. • Know what is meant by the term 'metadata' and be able to give examples. • Understand the effect of colour depth and resolution on the size of an image file. Understand how sound can be sampled and stored in digital form. • Understand how sampling intervals, sample size and bit rate affect the size of a sound file and the quality of its playback. Know why data is often compressed for transfer and storage. • Understand the difference between lossy and lossless compression. • Know why some types of data are only suitable for one type of compression. Know why data is often compressed for transfer and storage. • Understand the difference between lossy and lossless compression. • Know why some types of data are only suitable for one type of compression.				
Spring 2	Unit 1 (revisiting) 1.1Systems	Use of revision resources, past papers, & smart revise service from www.craigndave.org to revisit & recap content for Unit 1 Computer Systems.				Use of smart revise to identify areas for focus on individual basis, practice past papers &

	Architecture 1.2Memory 1.3Storage 1.4Wired & wireless networks 1.5Network topologies, protocols & layers 1.6System security 1.7System software 1.8Ethical, legal, cultural & environmental concerns	Use of smart revise to identify areas for focus on individual basis, practice past papers & exam techniques for focussed solutions. For content analysis of understanding & skills, please refer to year 10 long term plan for Computer Science GCSE, where Unit 1 was initially delivered.				exam techniques for focussed solutions.
Summer 1	Unit 2 (Revisiting) 2.1Algorithms 2.2Programming Techniques 2.3Producing Robust Programs	Use of revision resources, past papers, & smart revise service from www.craigndave.org to revisit & recap content for Unit 2 Computational Algorithms. Use of smart revise to identify areas for focus on individual basis, practice past papers & exam techniques for focussed solutions. For content analysis of understanding & skills, please refer to year 10 long term plan for Computer Science GCSE, where Unit				Use of smart revise to identify areas for focus on individual basis, practice past papers & exam techniques for focussed solutions

	2.4Computational Logic 2.5Translators & facilities of languages 2.6Data Representation	2 was initially delivered.				
Summer 2						