George Stephenson High School Secondary Scheme of Work: Year 10 BTEC Digital Information Technology

Unit	Lessons	Key Progression Indicators Knowledge, Understanding and Skills
Component 1: Learning Aim A: Exploring user interface design		A: Investigate user interface design for individuals and organisations A.2D1, A.2M1, A.2P1, A.2P2, A.1M1, A.1M2, A.1P1 and A.1P2 B: Use project planning techniques to plan and design a user interface B.2D2, B.2M2, B.2P3, B.2P4, B.1M3, B.1M4, B.1P3 and B.1P4 C: Develop and review a user interface C.2D3, C.2M2, C.2P5, C.2P6, C.1M5, C.1M6, C.1P5 and C.1P6 A: Investigate the role and impact of using data on individuals and organisations A.2D1, A.2M1, A.2P1, A.2P2, A.1M1, A.1M2, A.1P1 and A.1P2
Component 1: Learning Aim B : The Project Plan		
Component 1: Learning Aim C: Stadium Information System User Interface		
Component 2: Learning Aim A: Collecting, Presenting and Interpreting Data		
Component 2: Learning Aim B: Data Dashboard		
Component 3: Learning Aim A: Modern Technologies		

Year 10 BTEC Digital Information Technology Calendar

Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13
Component 1 Learning Aim A: Exploring user interface design												

Week 14	Week 15	Week 16	Week 17	Week 18	Week 19	Week 20	Week 21	Week 22	Week 23	Week 24	Week 25	Week 26
Component 1 Learning Aim B : The Project Plan				Component 1 Learning Aim C: Stadium Information System User Interface					Component 2: Learning Aim A: Collecting, Presenting and Interpreting Data			

Week 27	Week 28	Week 29	Week 30	Week 31	Week 32	Week 33	Week 34	Week 35	Week 36	Week 37	Week 38	Week 39
	Component 2: Learning Aim B: Data Dashboard											
	Component 3: Learning aim A: Modern Technologies											

George Stephenson High School Secondary Scheme of Work: Year 10 GCSE

Unit	Lessons	Key Progression Indicators Knowledge, Understanding and Skills

1.1 Systems Architecture	6	The unit is subdivided into five learning hours spread across six lessons including a test. The unit looks at the various components of the CPU used in the Von Neumann architecture. Students also investigate assembly language in the form of Little Man Computer.
		
		At the end of this Unit all students should be able to:
		• Understand the purpose of the CPU
		• Explain the role and operation of the following CPU registers used in Von Neumann architecture:
		  • MAR (Memory Address Register),
		  • MDR (Memory Data Register),
		  • Program Counter,
		  • Accumulator
		• Describe common CPU components and their function: ALU (Arithmetic Logic Unit), CU (Control Unit), Cache
		• Explain the function of the CPU as fetch and execute instructions stored in memory
		• Describe how common characteristics of CPUs affect their performance: clock speed, cache size, number of cores
		• Explain the purpose and give examples of embedded systems

1.2 Memory	6	The unit is subdivided into five learning hours spread across six lessons including a test. We build on the fundamentals learned in 1.1 now concentrating on RAM, ROM, cache and the need for virtual memory. **At the end of this Unit all students should be able to:** • Describe the difference between RAM and ROM • Describe the purpose RAM and ROM in a computer system • Explain the need for virtual memory • Describe flash memory

1.3 Storage	6	The unit is subdivided into five learning hours spread across six lessons including a test. The unit examines the need for secondary storage devices and their practical advantages in given applications. **At the end of this Unit all students should be able to:** • Discuss the need for secondary storage including optical, magnetic and solid state storage • Discuss data capacity of storage devices and • Calculate data capacity requirements • Evaluate suitable storage devices and storage media for a given application using the following characteristics: capacity, speed, portability, durability, reliability, cost
1.4 Wired and Wireless Networks	9	The unit is subdivided into eight learning hours spread across nine lessons, including a test. The unit begins by explaining the Internet and IP addressing, with practical exercises to help students understand the role of packet switching and DNS services. The lessons move on to look at LAN network topologies and Ethernet, with further material on virtual networking. Wireless networking, frequencies and encryption covered. **At the end of this Unit all students should be able to:** • Explain the advantages of networking stand-alone computers into a local area network • Explain the different roles of computers in a client-server and a peer-to-peer network • Describe the differences between a local area network and a wide area network such as the Internet • Describe the nature of the Internet as a worldwide collection of computer networks • Identify different transmission media • Describe the hardware needed to connect to the Internet including routers and switches • Explain the need for IP addressing of resources on the Internet and how this can be facilitated by the role of DNS servers • Explain the concept of encryption, giving examples • Explain how Wi-Fi frequencies and channels affect connectivity and transmission

1.8 Ethical, Legal, Cultural, Environmental concerns	12	The unit is subdivided into eleven learning hours and an end-of-unit test spread across twelve lessons, including a test. It covers different computer technologies and applications and the ethical, environmental and legal considerations surrounding them are described. **At the end of this Unit all students should be able to:** • Describe some ethical, legal, cultural and/or environmental issues in relation to a given scenario • Describe some privacy issues in relation to a given scenario • Describe the differences between open source and proprietary software and give advantages of each • List the clauses of the Data Protection Act and Computer Misuse Act and give examples of situations in which they are relevant • Evaluate the impact of and issues related to the use of computers in society
2.1 Algorithms	18	The unit is subdivided into eighteen lessons including a test. It is a theoretical unit, the initial lessons introduce the concepts of computational thinking; abstraction, decomposition and algorithmic thinking. Lessons on standard searching and sorting algorithms are next followed by lessons on developing algorithms using flow diagrams and pseudocode. The unit finishes with lessons on interpreting, correcting and completing algorithms. **At the end of this Unit all students should be able to:** • Explain how abstraction is used in a given scenario • Explain how decomposition may be used in an algorithm for a given problem • Explain how a binary search works • Explain how a bubble sort works • Explain how a merge sort and an insertion sort work • Use pseudocode to define the steps in a complex algorithm • Correct or complete a complex algorithm
2.2 Programming Techniques	24	The unit is subdivided into 24 lessons including a test. It covers data types and arithmetic operations. Sequence, selection and iteration are covered, followed by lessons on arrays, procedures and functions with lesson time given to understanding records and reading from and writing to a text file. This unit incorporates a large amount of practical programming and therefore this unit is split across the academic year as well as being given a continuous series of lessons. **At the end of this Unit all students should be able to:** • Identify and use variable types integer, real, Boolean, character and string

		• Identify variables and constants in a program • Use meaningful identifier names and know why it is important to use them • Use arithmetic operations including mod and div • Use random number generation • Follow through pseudocode solutions to simple problems involving sequence, selection and iteration • Explain why functions and procedures are used in creating solutions to problems • Write pseudocode solutions to simple problems involving sequence, selection and iteration • Use nested selection and iteration statements • Use Boolean operations NOT, AND and OR within conditions for iterative and selection structures • Use basic string manipulation functions in pseudocode solutions • Give examples of data structures: arrays and records • Use one-dimensional and two-dimensional arrays in the design of solutions to simple problems • Write simple functions and procedures using parameters • Read from and write to a text file
2.3 Producing Robust Programs	9	The unit is subdivided into 9 lessons including a test. In this unit students will learn what is meant by defensive design, and practise writing and coding algorithms which incorporate this. Different types of error, and how to detect them in a program code, is described along with testing and creating a test plan. **At the end of this Unit all students should be able to:** • Describe some simple validation checks that can be applied to data • Select test data that covers normal (typical), boundary (extreme) and erroneous data • Use a trace table to find errors or determine the purpose of an algorithm • Be able to justify the choice of test data
3.1 Programming Project	21	The students must take part in a 20-hour programming project. The students will be given a scenario to work on through which they must individually plan, design, develop, and evaluate a programming solution, to the scenario. Students will have acquired the relevant skills to complete this project during units 2.1, 2.2, and 2.3.

Year 10 GCSE Computing Calendar

Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13
1.1 Systems Architecture & 2.2 Programming Techniques			1.2 Memory & 2.2 Programming Techniques			1.3 Storage & 2.2 Programming Techniques			1.4 Wired and Wireless Networks			

Week 14	Week 15	Week 16	Week 17	Week 18	Week 19	Week 20	Week 21	Week 22	Week 23	Week 24	Week 25	Week 26
2.1 Algorithms						1.8 Ethical, Legal, Cultural, Environmental concerns				2.2 Programming Techniques		

Week 27	Week 28	Week 29	Week 30	Week 31	Week 32	Week 33	Week 34	Week 35	Week 36	Week 37	Week 38	Week 39
2.2 Programming Techniques		2.3 Producing Robust Programs			Project Preparation	Programming Project						Work Experience

George Stephenson High School Secondary Scheme of Work: Year 11 BTEC Digital Information Technology

Unit	Lessons	Key Progression Indicators Knowledge, Understanding and Skills
Component 2 Learning Aim A: All about data	14	A: Investigate the role and impact of using data on individuals and organisations A.2D1, A.2M1, A.2P1, A.2P2, A.1M1, A.1M2, A.1P1 and A.1P2 B: Create a dashboard using data manipulation tools B.2D2, B.2M2, B.2P3, B.2P4, B.1M3, B.1M4, B.1P3 and B.1P4 C: Draw conclusions and review data presentation methods C.2D3, C.2M2, C.2P5, C.2P6, C.1M5, C.1M6, C.1P5 and C.1P6
Component 2 Learning Aim B: B: The Quiz		
Component 2 Learning Aim C: How can I make it better?		
Component 3: Learning Aim B: Threats to digital systems (Cyber Security)	6	
Component 3 Learning Aim C: Responsible, legal and ethical use of data (Implications of digital systems)	7	
Component 3 Learning Aim D: Planning and communication in digital systems	5	

Year 11 BTEC Digital Information Technology Calendar

Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13
Component 3: Learning Aim B: Threats to digital systems (Cyber Security)						Component 3 Learning Aim C: Responsible, legal and ethical use of data (Implications of digital systems)						Mock Exam
							Component 3 Learning Aim D: Planning and communication in digital systems					

Week 14	Week 15	Week 16	Week 17	Week 18	Week 19	Week 20	Week 21	Week 22	Week 23	Week 24	Week 25	Week 26
Component 2 Learning Aim B: The Quiz										Component 2 Learning Aim C: How can I make it better?		
						GCSE exam 4th Feb PM						

Week 27	Week 28	Week 29	Week 30	Week 31	Week 32	Week 33	Week 34	Week 35	Week 36	Week 37	Week 38	Week 39
Component 2 Learning Aim C: How can I make it better? CONT												

George Stephenson High School Secondary Scheme of Work: Year 11 GCSE

Unit	Lessons	Key Progression Indicators Knowledge, Understanding and Skills
1.5 Network Topologies, Protocols, and Layers	12	The unit is subdivided into eleven learning hours, including a test. The unit begins by investigating Client-server networks and hosting describing common protocols and the concept of layers. **At the end of this Unit all students should be able to:** • Describe, using diagrams or otherwise, the star and mesh network topologies • Explain the terms IP addressing, MAC addressing, packet and protocols • Describe network policies such as acceptable use, disaster recovery, backup and archiving • Describe the advantages and disadvantages of star and mesh network topologies • Describe the concept of hosting and Cloud services • Describe the different layers in the TCP/IP protocol stack and the protocols used at each stage • Explain the advantages of layering in this context
1.6 System Security	15	The unit is subdivided into 13 learning hours, including a test. The lessons cover network threats and ways of identifying and preventing vulnerabilities. **At the end of this Unit all students should be able to:** • Describe briefly threats posed to networks including brute force attacks, denial of service attacks, data interception and theft, poor network policy • Explain briefly what is meant by phishing and how to keep data safe from phishing attacks • List precautions which can be taken to keep data safe from hackers including anti-malware software, firewalls, user access levels, passwords and encryption • Describe ways of identifying and preventing network vulnerabilities, including the use of passwords, encryption, penetration testing, network forensics and network policies • Explain what is meant by a social engineering attack and give examples • Explain what is meant by a Denial of Service attack and brute force attack • Explain the concept of SQL injection

1.7 System Software	9	The unit is subdivided into 13 learning hours, including a test. The lessons cover operating system and utility software. **At the end of this Unit all students should be able to:** • Describe the basic functions of an operating system: user interface, memory management, multi-tasking, peripheral management, user and file management • Explain briefly what is meant by memory management and multi-tasking • Describe briefly the purpose of encryption, defragmentation and data compression software • Describe different types of user interface • Describe utility system software: encryption software, defragmentation, data compression • Describe methods of backup (full and incremental) • Explain the need for the following functions of an operating system: memory management, peripheral management, multi-tasking and user management • Explain briefly why increasing the length of an encryption key increases the strength of encryption
2.4 Computational Logic	6	The unit is subdivided into 6 lessons, including a test. It concentrates on Boolean logic, logic diagrams and truth tables. **At the end of this Unit all students should be able to:** • Recognise standard symbols used to represent NOT, AND OR, NAND, NOR and XOR logic gates • Draw truth tables for the above logic gates • Recognise a logic gate from its truth table • Draw a logic circuit to solve a given problem
2.5 Translators and Facilities of Languages	6	The unit is subdivided into 6 lessons, including a test. The unit looks at the classification of programming languages and the different types of translator used with high- and low-level languages. **At the end of this Unit all students should be able to:** • Give examples of high-level and low-level languages • Give advantages of high-level languages over low-level languages • Explain the differences between a compiler, interpreter and assembler • Give examples and reasons of when it might be appropriate to use a low-level language • Give examples of when it would be appropriate to use a compiler and interpreter

2.6 Data Representation	12	The unit is subdivided into 12 lessons, including a test. This unit looks at units of information and the conversion of integers from decimal to binary and vice versa. Moving on to binary arithmetic, the use of hexadecimal numbers and the binary representation of characters. Check digits are also covered by practical example. Representation of images and sound, and compression techniques are also covered. **At the end of this Unit all students should be able to:** • Explain why all data needs to be converted to binary before the computer can process it • Convert positive denary whole numbers (0-255) into 8-bit binary numbers and vice versa • Convert between binary and hexadecimal • Explain the use of binary codes to represent characters • Understand the term 'character set' • Explain the relationship between the number of bits per character and the number of characters which can be represented • Explain the representation of an image as a series of pixels represented in binary • Explain how sound can be sampled and stored in digital form • Perform a binary shift • Explain the need for compression • Add two binary integers and explain overflow errors • Explain why hexadecimal numbers are used to represent binary data • Discuss the effect of colour depth and resolution on the size of an image file • Explain how sampling intervals and other considerations affect the size of a sound file • Explain the effects of a binary shift • Explain the purpose of a check digit • Explain the effect of different types of compression
Revisit and Revision	21	Time is given to revisit topics before the exam period. This time is also used to complete Mock Exams and for feedback to be given.

Year 11 GCSE Calendar

Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13
1.5 Network Topologies, Protocols and Layers				1.6 System Security					1.7 System Software		Mock Exam	1.7 System Software

Week 14	Week 15	Week 16	Week 17	Week 18	Week 19	Week 20	Week 21	Week 22	Week 23	Week 24	Week 25	Week 26
2.4 Computational Logic		2.5 Translators and Facilities of Languages		2.6 Data Representation				Revisit and Revision				

Week 27	Week 28	Week 29	Week 30	Week 31	Week 32	Week 33	Week 34	Week 35	Week 36	Week 37	Week 38	Week 39
Revisit and Revision		GCSE Computer Science Exam										