



Computing curriculum overview KS3

Computing curriculum overview – Year 7 (KS3)

Topic	7.1 ICT Skills	7.2 E-safety – Social networks	7.3 Computer systems – Computer hardware (csuk)	7.4 Coding with Scratch (csuk)	7.5 Game design in Scratch (assessment unit)
Length of topic (in weeks)	6	7	6	8	8
Links to Curriculum	Undertake creative projects that involve selecting, using, and combining multiple applications, preferably across a range of devices, to achieve challenging goals, including collecting and analysing data and meeting the needs of known users. Understand a range of ways to use technology safely, respectfully, responsibly and securely, including protecting their online identity and privacy; recognise inappropriate content, contact and conduct, and know how to report concerns.	understand a range of ways to use technology safely, respectfully, responsibly and securely, including protecting their online identity and privacy; recognise inappropriate content, contact and conduct, and know how to report concerns.	understand the hardware and software components that make up computer systems, and how they communicate with one another and with other systems understand how instructions are stored and executed within a computer system; understand how data of various types (including text, sounds and pictures) can be represented and manipulated digitally, in the form of binary digits.	use 2 or more programming languages, at least one of which is textual, to solve a variety of computational problems; make appropriate use of data structures [for example, lists, tables or arrays]; design and develop modular programs that use procedures or functions.	Use 2 or more programming languages, at least one of which is textual, to solve a variety of computational problems; make appropriate use of data structures [for example, lists, tables or arrays]; design and develop modular programs that use procedures or functions Undertake creative projects that involve selecting, using, and combining multiple applications, preferably across a range of devices, to achieve challenging goals, including collecting and analysing data and meeting the needs of known users. understand a range of ways to use technology safely, respectfully, responsibly and securely, including protecting their online identity and privacy; recognise inappropriate content, contact and conduct, and know how to report concerns.
Assessment Task(s)	Collection of documents that show development of digital literacy skills.	Demonstrate understanding of safety concepts through the production of digital artifacts.	Multiple choice test	Programming of a calculator using key programming concepts of sequencing, selection and iteration.	Design and create a multi level game. Unit is an assessment of previous units taught in the year.
Key Knowledge	Development of digital literacy understanding common features of programs. Understanding of hardware and how to use it.	Understanding social networks, personal data and privacy settings. Understanding and explaining cyberbullying and how to report it. Explain dangers to computers when online and how to avoid these. Intro to e-safety primary year 6 (external provider)	What is a computer Key components within a computer How key components work within the computer Understanding of the CPU. Basics of computers/what are computers year 5 and 6 (external provider)	Understand how to input & output values in scratch. Understand how to use variables, iteration and decisions in scratch. Understand how to break a problem down in to small steps Basic problem solving with scratch (external provider) Kirkby High School transition work year 6.	Understand how to input & output values in scratch. Understand how to use variables, iteration and decisions in scratch. Understand how to break a problem down in to small steps Basic problem solving with scratch (external provider) Kirkby High School transition work year 6. Understanding social networks, personal data and privacy settings. Understanding and explaining cyberbullying and how to report it. Explain dangers to computers when online and how to avoid these. Intro to e-safety primary year 6 (external provider)
Key Skills	General ICT skills	Knowledge and understanding of risks and methods to avoid risks. Identifying risks and dangers to computers. Digital literacy skills. Group work skills.	Identify what makes a computer system. Identify none standard computer systems and justify identification. Digital literacy skills	Decomposition, abstraction, computational thinking. Digital literacy Noticing details, independence, perseverance.	Decomposition, abstraction, computational thinking. Digital literacy Noticing details, independence, perseverance. Knowledge and understanding of risks and methods to avoid risks. Identifying risks and dangers to computers. Digital literacy skills.

Computing curriculum overview – Year 8 (KS3)

Topic	8.1 Control Systems with FLOWOL	8.2 First steps in Small Basic (sequence and iteration)	8.3 Binary Bits and Bobs	8.4 Coding with Microbit Madness	8.5 introduction to spreadsheets
Length of topic (in weeks)	6	8	7	6	8
Links to Curriculum	design, use and evaluate computational abstractions that model the state and behaviour of real-world problems and physical systems.	use 2 or more programming languages, at least one of which is textual, to solve a variety of computational problems; make appropriate use of data structures [for example, lists, tables or arrays]; design and develop modular programs that use procedures or functions	understand how instructions are stored and executed within a computer system; understand how data of various types (including text, sounds and pictures) can be represented and manipulated digitally, in the form of binary digits.	design, use and evaluate computational abstractions that model the state and behaviour of real-world problems and physical systems use 2 or more programming languages, at least one of which is textual, to solve a variety of computational problems; make appropriate use of data structures [for example, lists, tables or arrays]; design and develop modular programs that use procedures or functions.	understand a range of ways to use technology safely, respectfully, responsibly undertake creative projects that involve selecting, using, and combining multiple applications, preferably across a range of devices, to achieve challenging goals, including collecting and analysing data and meeting the needs of known users
Assessment Task(s)	Creation of a series of solutions to real world scenarios.	Students will complete an assessment portfolio at the end of the Unit. They will amend an existing program to create a screensaver, paste in evidence of their finished program and complete a brief self-assessment.	Multiple choice test	Production of a portfolio of solutions to given problems with written explanations of the solutions.	Production of portfolio of spreadsheets to show use of key skills. Answering of questions to show understanding of solutions provided.
Key Knowledge	Design, use, and evaluate computational abstractions that model the state and behaviour of real-world problems and physical systems.	•Use a textual programming language to solve a variety of computational problems; make appropriate use of data structures; design and develop modular programs that use procedures and functions •Use built in assistant to help develop code	•Understand how instructions are stored and executed within a computer system; understand how data of various types (including text, sounds, and pictures) can be represented and manipulated digitally, in the form of binary digits; be able to convert between binary and decimal, and perform simple binary arithmetic	What is a miniature computer Compiling Flashing Random number generation Using accelerometers and built in compass functions.	Format a simple spreadsheet Use simple formulae and functions Name cells in a spreadsheet Create a basic pie chart to display results
Key Skills	Decomposition, abstraction, computational thinking. Digital literacy Noticing details, independence, perseverance.	Decomposition, abstraction, computational thinking. Digital literacy Noticing details, independence, perseverance.	Show how numbers and text can be represented in binary. Binary addition, binary conversion. Digital literacy.	Decomposition, abstraction, computational thinking. Digital literacy Noticing details, independence, perseverance.	Digital literacy Formatting and presenting information
Enrichment opportunities: Robotics club (after school) February half term onwards IDEA award (introduced in school but then worked on in own time by students).					

Computing curriculum overview – Year 9 (KS3)

Topic	9.1 Computational thinking and logic	9.2 Next steps in Small Basic (iteration)	9.3 Computer Crime and cyber security	9.4 Spreadsheet models	9.5 Computer Systems – Understanding Computers and networks.
Length of topic (in weeks)	7	8	6	7	8
Links to curriculum	understand several key algorithms that reflect computational thinking [for example, ones for sorting and searching]; use logical reasoning to compare the utility of alternative algorithms for the same problem Understand simple Boolean logic [for example, AND, OR and NOT] and some of its uses in circuits and programming;	use 2 or more programming languages, at least one of which is textual, to solve a variety of computational problems; make appropriate use of data structures [for example, lists, tables or arrays]; design and develop modular programs that use procedures or functions	understand a range of ways to use technology safely, respectfully, responsibly and securely, including protecting their online identity and privacy; recognise inappropriate content, contact and conduct, and know how to report concerns	design, use and evaluate computational abstractions that model the state and behaviour of real-world problems and physical systems understand the hardware and software components that make up computer systems, and how they communicate with one another and with other systems	understand the hardware and software components that make up computer systems, and how they communicate with one another and with other systems understand how instructions are stored and executed within a computer system; understand how data of various types (including text, sounds and pictures) can be represented and manipulated digitally, in the form of binary digits understand how numbers can be represented in binary, and be able to carry out simple operations on binary numbers [for example, binary addition, and conversion between binary and decimal]
Assessment Task(s)	Exam style question Multiple choice test	Students will complete an assessment portfolio at the end of the Unit. They will amend an existing program to create a screensaver, paste in evidence of their finished program and complete a brief self-assessment.	Exam style question Multiple choice test	Assessment portfolio	Exam style question Multiple choice test
Key Knowledge	- understand simple Boolean logic [for example, AND, OR and NOT] and some of its uses in circuits and programming; - Be able to ask logical questions to solve problems. Know the common Boolean operators: AND, OR, NOT. Know different logic gates including: AND gates, OR gates, NOT gates Understand what an algorithm is Create a sequence of instructions to achieve a goal	- Use a textual programming language to solve a variety of computational problems; make appropriate use of data structures; design and develop modular programs that use procedures and functions - Use built in assistant to help develop code	understand a range of ways to use technology safely, respectfully, responsibly and securely, including protecting their online identity and privacy; recognise inappropriate content, contact and conduct and know how to report concerns	Give examples of how computer models are used in the real world Format a simple spreadsheet model Use simple formulae and functions Name cells in a spreadsheet model Use a simple spreadsheet model to explore different “what if” scenarios Create a basic pie chart to display results	- Understand the hardware and software components that make up computer systems, and how they communicate with one another and with other systems - Understand how instructions are stored and executed within a computer system; be able to convert between binary and decimal, and perform simple binary arithmetic
Key Skills	Digital literacy Research Reading to understand Reasoning Adapting Independence	Decomposition, abstraction, computational thinking. Digital literacy Noticing details, independence, perseverance.	Digital literacy Research Reading to understand	Digital literacy Research Reading to understand Reasoning	Digital literacy Research Using number bases

Enrichment opportunities:

January onwards IDEA award (introduced in school but then worked on in own time by students)



Computing curriculum overview KS4

Computing curriculum overview – Year 10 (KS4) Exam board: OCR

Topic	1.1 Systems Architecture	Practical programming	1.2 Memory and Storage (part 1)	Practical programming	1.2 Memory and Storage (part 2)	Practical programming	1.3 – Computer networks, connections and protocols	Practical programming
Length of topic (in weeks)	2 (6 Lessons)		2 (5 Lessons)		4 (12 Lessons)		5 (14 lessons)	
Links to specification	1.1.1 Architecture of the CPU 1.1.2 CPU performance 1.1.3 Embedded systems		1.2.1 Primary storage (Memory) 1.2.2 Secondary storage		1.2.3 Units 1.2.4 Data storage 1.2.5 Compression		1.3.1 Networks and topologies 1.3.2 Wired and wireless networks, protocols and layers	
	develop their capability, creativity and knowledge in computer science		develop their capability, creativity and knowledge in computer science		develop their capability, creativity and knowledge in computer science		develop their capability, creativity and knowledge in computer science	
Assessment Task(s)	SLR 1.1 Student workbook SLR 1.1 End of topic test		SLR 1.2 Student workbook (part 1) SLR 1.2 End of topic test (part 1)		SLR 1.2 Student workbook (part 2) SLR 1.2 End of topic test (part 2)		SLR 1.3 Student workbook SLR 1.3 End of topic test	
Key Knowledge	- Understand the CPU: registers, factors affecting speed. - Know the stages of the fetch, execute cycle. - Describe the Von Neumann architecture and it's components.		- Understand primary storage: RAM and ROM, Virtual memory - Understand the need for secondary storage including types of storage, suitability of storage devices, advantages and disadvantages		- Understand units of data storage, representing data capacity, converting between units. - Understand computers use binary - Know how to convert positive denary whole numbers to binary numbers and vice versa, adding two binary integers together and explain overflow errors - Know how to convert positive numbers between denary, binary and 2 digit hexadecimal - Know how to apply Binary shifts - Understand how computers store Numbers, Characters, Images and Sound using binary. - Understand compression.		- Know different types of network: LAN WAN - Know factors that affect the performance of networks - Know different roles of computers in a client-server and a peer-to-peer network - Know the hardware needed for a Local Area Network - Understand the Internet - Know Star and Mesh network topologies - Understand modes of connection: Wired Wireless - Understand encryption - Understand IP and MAC addressing - Understand the need for standards and common protocols including: The concept of layers	
Key Skills	Demonstrate knowledge and understanding of the key concepts and principles of Computer Science.		Demonstrate knowledge and understanding of the key concepts and principles of Computer Science.		Demonstrate and apply knowledge and understanding of key concepts and principles of Computer Science.		Demonstrate and apply knowledge and understanding of key concepts and principles of Computer Science.	

Computing curriculum overview – Year 10 (KS4) Exam board: OCR

Topic	1.4 – Network security	Practical programming	1.5 – Systems software	Practical programming	1.6 – Ethical, legal, cultural and environmental concerns	Practical programming	2.2 Programming fundamentals	Practical programming	Practical Programming
Length of topic (in weeks)	4 (12 Lessons)		2 (6 Lessons)		3 (9 lessons)		3 (9 lessons)		Continuous throughout the year. Text adventure game 3 weeks end of year
Links to specification	1.4.1 Threats to computer systems and networks 1.4.2 Identifying and preventing vulnerabilities		1.5.1 Operating systems 1.5.2 Utility software		1.6.1 from the GCSE specification (J277)		2.2.1 Programming fundamentals 2.2.2 Data types 2.2.3 Additional programming techniques		Practical Programming
	develop their capability, creativity and knowledge in computer science understand how changes in technology affect safety, including new ways to protect their online privacy and identity, and how to report a range of concerns		develop their capability, creativity and knowledge in computer science		develop their capability, creativity and knowledge in computer science understand how changes in technology affect safety, including new ways to protect their online privacy and identity, and how to report a range of concerns		develop and apply their analytic, problem-solving, design, and computational thinking skills		develop and apply their analytic, problem-solving, design, and computational thinking skills
Assessment Tasks	SLR 1.4 Student workbook SLR 1.4 End of topic test		SLR 1.5 Student workbook SLR 1.5 End of topic test		SLR 1.6 Student workbook SLR 1.6 End of topic test		SLR 2.2 Student workbook SLR 2.2 End of topic test		Programming portfolio
Key Knowledge	•Understand threats posed to devices/systems •Knowledge/principles of each form of attack including: How the attack is used and the purpose of the attack •Understanding of how to limit the threats posed in 1.4.1 •Understanding of methods to remove vulnerabilities •Knowledge/principles of each prevention method: What each prevention method may limit/ prevent How it limits the attack		•Understand the purpose and functionality of operating systems •Know what multitasking is •Understand what the OS manages and the need for device drivers. •Understand utility software		•Understand ethical issues of computer development •Understand privacy issues of computer technologies. •Understand legal issues of computer technologies. •Understand cultural issues of computer technologies. •Understand environmental issues of computer technologies. •Understand how digital technologies impact society. •Know and understand the difference between Open source vs proprietary software.		•Know the basic programming constructs and key terms. •Understand data types, operators and string manipulation •Understand how to use basic file handling operations. •Understand data structures such as: Records and SQL, arrays and sub problems. •Know how to use procedures and functions within programs. •Understand how to use random number generation.		Use of a high level programming language to solve a variety of computational problems; make appropriate use of data structures; design and develop modular programs that use procedures and functions
Key Skills	Apply knowledge and understanding of key concepts and principles of Computer Science.		Apply knowledge and understanding of key concepts and principles of Computer Science.		Demonstrate knowledge and understanding of the key concepts and principles of Computer Science.		Apply knowledge and understanding of key concepts and principles of Computer Science		Design, write, test and refine computer programs Analyse problems in computational

Computing curriculum overview – Year 11 (KS4) Exam board: OCR

Topic	1.3 – Computer networks, connections and protocols	1.5 – Systems software	Python Programming	2.1 Algorithms
Length of topic (in weeks)	5 (14 lessons)	2 (6 Lessons)	19 Lessons	2 (5 lessons) + 2 (exam revision weeks)
Links to specification	1.3.1 Networks and topologies 1.3.2 Wired and wireless networks, protocols and layers	1.5.1 Operating systems 1.5.2 Utility software	Practical Programming skills	2.1.1 – 2.1.3 from the GCSE specification (J277)
	develop their capability, creativity and knowledge in computer science understand how changes in technology affect safety, including new ways to protect their online privacy and identity, and how to report a range of concerns	develop their capability, creativity and knowledge in computer science understand how changes in technology affect safety, including new ways to protect their online privacy and identity, and how to report a range of concerns	develop and apply their analytic, problem-solving, design, and computational thinking skills	develop and apply their analytic, problem-solving, design, and computational thinking skills
Assessment Task(s)	SLR 1.3 Student workbook SLR 1.3 End of topic test	SLR 1.5 Student workbook SLR 1.5 End of topic test	Continuous throughout the unit	SLR 2.1 Student workbook SLR 2.1 End of topic test Exam revision folder
Key Knowledge	•Know different types of network: LAN WAN •Know factors that affect the performance of networks •Know different roles of computers in a client-server and a peer-to-peer network •Know the hardware needed for a Local Area Network •Understand the Internet •Know Star and Mesh network topologies •Understand modes of connection: Wired Wireless •Understand encryption •Understand IP and MAC addressing •Understand the need for standards and common protocols including: The concept of layers	•Understand the purpose and functionality of operating systems •Know what multitasking is •Understand what the OS manages and the need for device drivers. •Understand utility software	This unit teaches students how to code, logical thinking skills focused on syntax structure and the 3 key areas of coding. 1. Sequence 2. Selection 3. Iteration	•Know what is meant by abstraction and give examples. •Understand decomposition and its use in programming •Understand using structure diagrams for decomposition •Understand how to solve computational problems using decomposition. Exam Revision Gain experience of answering computational thinking, algorithms and programming questions for component J277/02 using exam revision unit. (5 Lessons)
Key Skills	Demonstrate and apply knowledge and understanding of key concepts and principles of	Apply knowledge and understanding of key concepts and principles of Computer	Be able to write and debug Python Programs	Demonstrate and apply knowledge and understanding of key concepts and principles of

Computing curriculum overview – Year 11 (KS4) Exam board: OCR

Topic	2.3 Producing robust programs		2.4 Boolean logic		2.5 Programming languages and IDEs		Revision
Length of topic (in weeks)	3 (8 lessons) + 3 (exam revision weeks)		1 (3 lessons) + 1 (exam revision week)		2 (4 lessons)		6
Links to specification	2.3.1 – 2.3.2 from the GCSE specification (J277)		2.4.1 from the GCSE specification (J277)		2.5.1 – 2.5.2 from the GCSE specification (J277)		
	develop and apply their analytic, problem-solving, design, and computational thinking skills		develop and apply their analytic, problem-solving, design, and computational thinking skills		develop and apply their analytic, problem-solving, design, and computational thinking skills		
Assessment Task(s)	SLR 2.3 Student workbook SLR 2.3 End of topic test Exam revision folder		SLR 2.4 Student workbook SLR 2.4 End of topic test Exam revision folder		SLR 2.5 Student workbook SLR 2.5 End of topic test Exam revision folder		
Key Knowledge	• Know what is meant by “defensive design considerations” including validation and potential problems that can occur. • Understand the importance of maintainability and refining algorithms • Understand testing and test data	Exam Revision Gain experience of answering computational thinking, algorithms and programming questions for component J277/02 using exam revision unit. (7 Lessons)	• Know how to make simple logic diagrams • Understand how to complete truth tables • Understand how to create, complete or edit logic diagrams and truth tables for given scenarios.	Exam Revision Gain experience of answering computational thinking, algorithms and programming questions for component J277/02 using exam revision unit. (2 Lessons)	• Know characteristics of high level programming languages • Understand key terms: source code assembly code machine code. • Understand low level programming by writing programs in low level language. • Know what a translator does • Understand the different types of translator. • Know what an IDE is and a range of key features provided by them.	Exam Revision Gain experience of answering computational thinking, algorithms and programming questions for component J277/02 using exam revision unit. (3 Lessons)	
Key Skills	Demonstrate and apply knowledge and understanding of key concepts and principles of Computer Science. Design, write, test and refine		Demonstrate and Apply knowledge and understanding of key concepts and principles of Computer Science Analyse problems in computational terms:		Demonstrate and Apply knowledge and understanding of key concepts and principles of Computer Science.		



Cambridge Nationals IT curriculum overview KS4

Computing iMedia curriculum overview – Year 10 (KS4) Exam board: OCR

Topic	R050
Length of topic (in weeks)	8 weeks
Links to specification	Topic Area 1: Design tools Topic Area 2: Human Computer Interface (HCI) in everyday life Topic Area 3: Data and testing
Assessment Task(s)	Knowledge test of Topic area 1, 2 and 3.
	develop and apply their analytic, problem-solving, design, and computational thinking skills
Key Knowledge	Know the components of each design tool • Know the type of software that can be used to create each design tool • Advantages and disadvantages of each design tool • Creating an original document using relevant design tools either using software or by sketches • Assessing the suitability of an original document to a given context • Know the purpose of HCI • Know why HCI is used for each application area • Know the importance of HCI applied to each application area • Advantages and disadvantages of the use of an HCI for each application area • Know the different display types and sizes that an HCI can be used on • The impact of display and resources on the HCI • Advantages and disadvantages of hardware considerations for using an HCI • Know how the HCI used on the Windows, Apple macOS, Apple's iOS, Android, Chrome, Ubuntu, Linux and Unix operating systems and software applications will impact on the design • Know how the HCI used on the digital platform will impact on the design • Know how a user will interact with the HCI • Advantages and disadvantages of each user interaction method • Know the difference between data and information • How data is converted to information • Know the characteristics of each data type • How each data type can be used • Assess the suitability and justify the use of data types applied to a given context • Alphanumeric is a combination of letters and numbers • Know the purposes of validation and verification • The different roles of validation and verification • The purpose of a data validation tool • How a validation tool can reduce user errors • The purpose of a data verification tool • How a verification tool can reduce user errors • Know the purpose of each data collection method • Advantages and disadvantages of the data collection methods • Assess the suitability and justify the use of the data collection methods applied to a given context • Advantages and disadvantages of each storage location • Know the characteristics of each storage device • Advantages and disadvantages of each storage device • Know why testing is needed • The effects of not testing on the final product • Advantages and disadvantages of testing • Know what each type of test data is • The role of each type of test data during testing • Advantages and disadvantages of each type of testing • Know what technical testing is • Know what tests can be used in technical testing • Know what user testing is • Know what tests can be used in user testing •
Key Skills	Students will learn the theoretical knowledge and understanding to apply design tools for applications, principles of human computer interfaces and the use of data and testing in different contexts when creating IT solutions or products.

Computing iMedia curriculum overview – Year 10 (KS4) Exam board: OCR

Topic	R060 (skill building)
Length of topic (in weeks)	6 weeks
Links to specification	Topic Area 2: Creating the spreadsheet solution 2.1 Use spreadsheet tools and techniques to create the solution
Assessment Task(s)	Generation of basic spreadsheets
	develop and apply their analytic, problem-solving, design, and computational thinking skills develop their capability, creativity and knowledge in computer science, digital media and information technology
Key Knowledge	Create a spreadsheet solution that is fit for purpose •Meaningful worksheet names •Manipulation of data using formulas and functions •Built in functions including SUM, MIN, MAX, AVERAGE, COUNT, IF, COUNTIF, •Solving formula errors (#DIV/0, #NAME?, #REF! etc) •Effective validation checks within the spreadsheet solution •worksheet editing •Use of different cell formatting options such as alignment, border, font, shading, text wrap and currency
Key Skills	Students will learn the basic key skills required for spreadsheet creation.

Computing iMedia curriculum overview – Year 10 (KS4) Exam board: OCR

Topic	R060 Creating spreadsheets
Length of topic (in weeks)	8 weeks
Links to specification	Topic Area 1: Planning and designing the spreadsheet solution Topic Area 2: Creating the spreadsheet solution
Assessment Task(s)	Knowledge tests Practice assessment
	develop and apply their analytic, problem-solving, design, and computational thinking skills develop their capability, creativity and knowledge in computer science, digital media and information technology
Key Knowledge	Produce design documents to create the spreadsheet solution including: o Functionality o Navigation system o Outputs from the system •Selection and use of appropriate software tools and techniques to effectively plan the spreadsheet solution • Design the functionalities for the solution •Design the calculations using flowcharts to enable others to understand calculations taking place •Design meaningful messages to be displayed to end users when errors occur • Be familiar with the creation of different types of outputs to meet user/client needs •Design different types of outputs to meet user/ client needs using visualisation diagrams and wireframes •Consideration of page layouts properties such as page size, print area, margins, headers/footers, guidelines, orientation and scaling •Consideration of house style/branding, colours, fonts, font styles, font size, alignment, logos/ images, cell formatting, chart formatting and labelling •Reports to present information to the client and the end user, consider where the information is coming from • Design a clear navigation system that meets the user/client needs using visualisation diagram(s) and/or wireframe(s) •The start-up and flow through the navigation system and being able to navigate back to the main menu •Show consideration of learnability and memorability in the design of the solution navigation •Accessibility considerations of sufficient contrast of text and colours, using meaningful names, screen tips •Layout considerations of use of white space, alignment, location of navigation tools on the user interface • Create a spreadsheet solution that is fit for purpose •Meaningful worksheet names •Manipulation of data using formulas and functions •Built in functions including SUM, MIN, MAX, AVERAGE, COUNT, IF, COUNTIF, LOOKUP, VLOOKUP, HLOOKUP, AND, OR, DATE, TODAY, SUMIF, SUBTOTAL •Relational operators including =, <, >, <=, >=, <> •Solving formula errors (#DIV/0, #NAME?, #REF! etc) •Effective validation checks within the spreadsheet solution •Naming of cells or a group of cells •Use of appropriate security measures such as lock cells, password protected workbook, worksheet editing •Use of different cell formatting options such as alignment, border, font, shading, text wrap and currency matching the plan and design of the solution •Modelling tools such as what-if and goal seek to predict different outcomes
Key Skills	In this unit you will learn the skills to be able to plan and design a spreadsheet solution to meet client requirements. You will be able to use a range of tools and techniques to create a spreadsheet solution based on your design, which you will test. You will be able to evaluate your solution based on the user requirements.

Computing iMedia curriculum overview – Year 10 (KS4) Exam board: OCR

Topic	R060 Creating spreadsheets
Length of topic (in weeks)	4 weeks
Links to specification	Topic Area 3: Testing the spreadsheet solution Topic Area 4: Evaluating the spreadsheet solution
Assessment Task(s)	Knowledge tests Practice assessment
	develop and apply their analytic, problem-solving, design, and computational thinking skills develop their capability, creativity and knowledge in computer science, digital media and information technology
Key Knowledge	Following a given test plan document which includes test number, test description, test data, expected result, actual result, remedial action, retesting •Choose appropriate test data to be used in the test plan •Technical testing: o Navigation features o Spreadsheet calculations o Content included in the output •How to record test results •How and when to retest • How suitable the spreadsheet solution is for the requirements of a client •How effectively the visual style has been generated •Has house style been followed •Whether the planned spreadsheet solution has been created •How the navigation system has successfully met the client's requirements Does not include: •User feedback / focus group review / user testing / user acceptance testing
Key Skills	In this unit you will learn the skills to be able to plan and design a spreadsheet solution to meet client requirements. You will be able to use a range of tools and techniques to create a spreadsheet solution based on your design, which you will test. You will be able to evaluate your solution based on the user requirements.