

Subject: GCSE Computer Science Year 10 Ability Mixed

Term / Date(s)	Component 2.4 (Strand 1), 2.1, 2.2 (Strand 2)	Component 1.6 (Strand 1), 2.1, 2.2 (Strand 2)	Component 2.5 (Strand 1), 2.1, 2.2 (Strand 2)
Topic	Strand 1 - Boolean logic Strand 2 - Programming	Strand 1 - Ethical, legal, cultural and environmental impacts Strand 2 - Programming	Strand 1 - Languages and Integrated Development Environments (IDE) Strand 2 - Programming
Topic overview	Strand 1 - Students will develop internal circuits to model how transistors in the CPU combine states to process binary data, as well as apply Boolean logic to operations within programming.	Strand 1 - Students will investigate how technology impacts upon society, including legislation, cultural, ethical and environmental issues. By exploring these issues, students will be able to respond to wider issues where technology may have positive and negative impacts.	Strand 1 - Students will explore the relationship between low-level languages (used by the CPU) and high-level languages used by programmers to develop software. Through this, students will investigate IDEs used to program and investigate common tools within them.
Pupils will learn...	Strand 2 - Students will develop and implement algorithms to solve a set of computational problems, debug errors and refine solutions.	Strand 2 - Students will develop and implement algorithms to solve a set of computational problems, debug errors and refine solutions.	Strand 2 - Students will develop and implement algorithms to solve a set of computational problems, debug errors and refine solutions.
Components	<p>Strand 1</p> <ul style="list-style-type: none"> Identify that Computers are made up of transistors that use switches in 2 states to function (binary, 1 and 0) Describe the purpose of AND, OR and NOT gates Draw in diagram form, each logic gate Describe the rules that each gate follows Use the diagram and rules to create a truth table that shows all possible outcomes of the gate Combine logic gates to create more complex logic gates with more possible outcomes Write Boolean algebraic expressions and recognise the use of symbols in expressions Use logic circuits as a solution to a range of potential computational problems <p>Strand 2</p> <ul style="list-style-type: none"> Describe and use abstraction and decomposition when defining, and refining problems / creating solutions to identify how to break problems down into smaller, manageable parts with relevant detail, leading to more effective solutions Recognise the purpose of shapes used within flowcharts to present algorithms to read solutions and predict outcomes and possible errors Create algorithms using flowcharts and a high-level programming language (Python) to solve a set problem, correct a solution or improve it Identify and use variables, constants, operators, inputs, outputs, common arithmetic operators, assignment operators and assignments in order determine what data should be input into a solution, what is done with the data and what the outcome should be Understand the need for sequencing within algorithms to ensure steps take place in the correct order, and recognise / fix solutions when the order is incorrect Understand and use selection within algorithms to make decisions and determine the next step within a solution, to design solutions that can adapt to the data inputted 	<p>Strand 1</p> <ul style="list-style-type: none"> Understand that technology can create ethical, legal, cultural, environmental and privacy issues. Identify a range of issues according to the categories above and discuss the impacts they have on society (both positive and negative). Be able to identify specific legislation and how it applies to individuals and larger groups (such as institutions, companies and countries). The legislation is: <ul style="list-style-type: none"> The Data Protection Act 2018 Computer Misuse Act 1990 Copyright, Designs and Patents Act 1988 Software Licenses (Open Source and Proprietary) Understand the use of software licencing and the different features and uses for open source and proprietary licencing. <p>Strand 2</p> <ul style="list-style-type: none"> Describe and use abstraction and decomposition when defining, and refining problems / creating solutions to identify how to break problems down into smaller, manageable parts with relevant detail, leading to more effective solutions Recognise the purpose of shapes used within flowcharts to present algorithms to read solutions and predict outcomes and possible errors Create algorithms using flowcharts and a high-level programming language (Python) to solve a set problem, correct a solution or improve it Identify and use variables, constants, operators, inputs, outputs, common arithmetic operators, assignment operators and assignments in order determine what data should be input into a solution, what is done with the data and what the outcome should be Understand the need for sequencing within algorithms to ensure steps take place in the correct order, and recognise / fix solutions when the order is incorrect Understand and use selection within algorithms to make decisions and determine the next step within a solution, to design solutions that can adapt to the data inputted 	<p>Strand 1</p> <ul style="list-style-type: none"> Identify examples of high and low-level languages. Understand the need for and differences between high and low-level languages. Describe characteristics of specific high-level and low-level languages such as Python and assembly code. Recognise the need to translate between high- and low-level languages. Describe the differences between compilers and interpreters, including the benefits and drawbacks of each. Recognise the common tools used within IDEs (integrated Development Environments) including: <ul style="list-style-type: none"> Editors Error diagnostics Run-time environment Translators <p>Strand 2</p> <ul style="list-style-type: none"> Describe and use abstraction and decomposition when defining, and refining problems / creating solutions to identify how to break problems down into smaller, manageable parts with relevant detail, leading to more effective solutions Recognise the purpose of shapes used within flowcharts to present algorithms to read solutions and predict outcomes and possible errors Create algorithms using flowcharts and a high-level programming language (Python) to solve a set problem, correct a solution or improve it Identify and use variables, constants, operators, inputs, outputs, common arithmetic operators, assignment operators and assignments in order determine what data should be input into a solution, what is done with the data and what the outcome should be Understand the need for sequencing within algorithms to ensure steps take place in the correct order, and recognise / fix solutions when the order is incorrect Understand and use selection within algorithms to make decisions and determine the next step within a solution, to design solutions that can adapt to the data inputted Understand and use iteration (count-controlled and condition-controlled) to repeat patterns within algorithms and design solutions efficiently

	<ul style="list-style-type: none"> Understand and use iteration (count-controlled and condition-controlled) to repeat patterns within algorithms and design solutions efficiently Identify the use of integer, real, Boolean, character/string and casting to choose or change to the correct data type needed for a specific scenario Explore the use of imported modules to perform specific tasks within an algorithm such as random value generation, drawing or use of time 	<ul style="list-style-type: none"> Understand and use iteration (count-controlled and condition-controlled) to repeat patterns within algorithms and design solutions efficiently Identify the use of integer, real, Boolean, character/string and casting to choose or change to the correct data type needed for a specific scenario Explore the use of imported modules to perform specific tasks within an algorithm such as random value generation, drawing or use of time 	<ul style="list-style-type: none"> Identify the use of integer, real, Boolean, character/string and casting to choose or change to the correct data type needed for a specific scenario Explore the use of imported modules to perform specific tasks within an algorithm such as random value generation, drawing or use of time
What pupils should already know (prior learning components)	<p>Strand 1 Within Key Stage 3 Computing, students will have been taught to</p> <ul style="list-style-type: none"> Recognise the need for logic gates to solve simple problems, and the use of specific logic gates and truth tables (<i>Year 7 – Binary and Logic</i>) <p>Strand 2 Within Key Stage 3 Computing, students will have been taught to</p> <ul style="list-style-type: none"> Recognise the use of shapes within flowcharts and apply them to design a solution to a real-world problem (<i>Year 9 – Algorithms</i>) Understand the meaning of abstraction and decomposition and apply them to break down or simplify problems (<i>Year 9 – Algorithms</i>) Use a programming language to solve several problems, and use a range data types and operators to perform arithmetic and logical operations (<i>Year 7 – Programming in Scratch, Year 8/9 – Python programming – Sequence</i>) Use logical operators and selection to determine decisions based on inputted data (<i>Year 7 / 8– Introducing Spreadsheets / Advanced Spreadsheets, Year 8/9 – Python programming – Sequence</i>) Recognise the use of specific programming syntax (rules) that must be followed when using a specific language <p>Within GCSE Compute Science, students will have been taught to</p> <ul style="list-style-type: none"> Use a high-level programming language to design programs to solve a specific problem Use specific data types to create variables Use sequencing, selection and iteration in combination to determine outcomes 	<p>Strand 1 Within Key Stage 3 Computing, students will have been taught to</p> <ul style="list-style-type: none"> Identify safe working practices and keep personal data secure (<i>Year 7 – Getting Started</i>). Understand the difference between genuine and fraudulent behaviour online, including phishing, pharming and other illegal behaviours (<i>Year 8 – Cyber Security</i>) Recognise the difference between ethical and legal issues, understand the meaning of key legislation and the impact of the Data Protection Act on how companies store and use data about individuals, and identify environmental impacts on society (<i>Year 9 – The Ethics of Computing</i>) <p>Strand 2 Within Key Stage 3 Computing, students will have been taught to</p> <ul style="list-style-type: none"> Recognise the use of shapes within flowcharts and apply them to design a solution to a real-world problem (<i>Year 9 – Algorithms</i>) Understand the meaning of abstraction and decomposition and apply them to break down or simplify problems (<i>Year 9 – Algorithms</i>) Use a programming language to solve several problems, and use a range data types and operators to perform arithmetic and logical operations (<i>Year 7 – Programming in Scratch, Year 8/9 – Python programming – Sequence</i>) Use logical operators and selection to determine decisions based on inputted data (<i>Year 7 / 8– Introducing Spreadsheets / Advanced Spreadsheets, Year 8/9 – Python programming – Sequence</i>) Recognise the use of specific programming syntax (rules) that must be followed when using a specific language <p>Within GCSE Compute Science, students will have been taught to</p> <ul style="list-style-type: none"> Use a high-level programming language to design programs to solve a specific problem Use specific data types to create variables Use sequencing, selection and iteration in combination to determine outcomes 	<p>Strand 1 Within Key Stage 3 Computing, students will have been taught to</p> <ul style="list-style-type: none"> Recognise the use of and features of a programming environment (<i>Year 7 – Programming Project / Year 9/9 - Python Programming</i>). <p>Strand 2 Within Key Stage 3 Computing, students will have been taught to</p> <ul style="list-style-type: none"> Recognise the use of shapes within flowcharts and apply them to design a solution to a real-world problem (<i>Year 9 – Algorithms</i>) Understand the meaning of abstraction and decomposition and apply them to break down or simplify problems (<i>Year 9 – Algorithms</i>) Use a programming language to solve several problems, and use a range data types and operators to perform arithmetic and logical operations (<i>Year 7 – Programming in Scratch, Year 8/9 – Python programming – Sequence</i>) Use logical operators and selection to determine decisions based on inputted data (<i>Year 7 / 8– Introducing Spreadsheets / Advanced Spreadsheets, Year 8/9 – Python programming – Sequence</i>) Recognise the use of specific programming syntax (rules) that must be followed when using a specific language <p>Within GCSE Compute Science, students will have been taught to</p> <ul style="list-style-type: none"> Use a high-level programming language to design programs to solve a specific problem Use specific data types to create variables Use sequencing, selection and iteration in combination to determine outcomes
Transferrable knowledge (skills)	<p>Strand 1</p> <ul style="list-style-type: none"> The understanding that all data processed by a computer is ultimately comprised of switches in two states (binary) and that all data must be converted to this <p>Strand 2</p> <ul style="list-style-type: none"> The use of a text-based programming language and its specific syntax to perform sequencing, selection and iteration Being able to articulate the need for programming constructs such as sequence, selection and iteration Being able to recognise the four main data types within programming 	<p>Strand 1</p> <ul style="list-style-type: none"> The ability to relate legislation to a wide range of contexts in which technology is used. The understanding that using information from the internet comes with potential legal implications <p>Strand 2</p> <ul style="list-style-type: none"> The use of a text-based programming language and its specific syntax to perform sequencing, selection and iteration Being able to articulate the need for programming constructs such as sequence, selection and iteration 	<p>Strand 1</p> <ul style="list-style-type: none"> The understanding that software comes with built-in tools that make use easier for designers Being able to recognise and describe common features across software tools and programming software <p>Strand 2</p> <ul style="list-style-type: none"> The use of a text-based programming language and its specific syntax to perform sequencing, selection and iteration Being able to articulate the need for programming constructs such as sequence, selection and iteration

	<ul style="list-style-type: none"> Being able to use a combination of mathematical operators to perform arithmetic and logical operations within the text-based programming language which solve a specific problem Being able to use exemplar code snippets and explain the outcome, as well as modify the code to perform actions needed for a specific purpose <p>General skills</p> <ul style="list-style-type: none"> The ability to search for information sources online and curate material based on relevance, factual content and needs of a specific purpose and audience Being able to use multiple pieces of software (such as a web browser, presentation software, image editing software and a cloud computing system) in quick succession to create and refine design projects The use of inference and articulation to obtain key knowledge from a topic and apply understanding when presenting findings 	<ul style="list-style-type: none"> Being able to recognise the four main data types within programming Being able to use a combination of mathematical operators to perform arithmetic and logical operations within the text-based programming language which solve a specific problem Being able to use exemplar code snippets and explain the outcome, as well as modify the code to perform actions needed for a specific purpose <p>General skills</p> <ul style="list-style-type: none"> The ability to search for information sources online and curate material based on relevance, factual content and needs of a specific purpose and audience Being able to use multiple pieces of software (such as a web browser, presentation software, image editing software and a cloud computing system) in quick succession to create and refine design projects The use of inference and articulation to obtain key knowledge from a topic and apply understanding when presenting findings 	<ul style="list-style-type: none"> Being able to recognise the four main data types within programming Being able to use a combination of mathematical operators to perform arithmetic and logical operations within the text-based programming language which solve a specific problem Being able to use exemplar code snippets and explain the outcome, as well as modify the code to perform actions needed for a specific purpose <p>General skills</p> <ul style="list-style-type: none"> The ability to search for information sources online and curate material based on relevance, factual content and needs of a specific purpose and audience Being able to use multiple pieces of software (such as a web browser, presentation software, image editing software and a cloud computing system) in quick succession to create and refine design projects The use of inference and articulation to obtain key knowledge from a topic and apply understanding when presenting findings
Key vocabulary pupil will know and learn	<p>Strand 1 - Input and Output, Switch, Transistor, Logic gate (AND gate, OR gate, NOT gate), Truth table, Inverter, Boolean Operation</p> <p>Strand 2 - Abstraction, Decomposition, pattern recognition, algorithm, Variables, constants, operators, inputs, outputs, selection, sequence, iteration, string, integer, float (decimal), Boolean operators, modules</p>	<p>Strand 1 – Privacy, Legal, Ethics, Copyright, Data Protection, Hacking Proprietary, Open Source, Digital Divide, E-Waste, Stakeholder</p> <p>Strand 2 - Abstraction, Decomposition, pattern recognition, algorithm, Variables, constants, operators, inputs, outputs, selection, sequence, iteration, string, integer, float (decimal), Boolean operators, modules</p>	<p>Strand 1 - Low-level language, High-level language, Generations (first, second, third), Machine code, Syntax, Translator, Compiler, Interpreter, Device driver, Debug, Source code, Integrated Development Environment, Text editor, Error diagnostics, Run-time environment</p> <p>Strand 2 - Abstraction, Decomposition, pattern recognition, algorithm, Variables, constants, operators, inputs, outputs, selection, sequence, iteration, string, integer, float (decimal), Boolean operators, modules</p>
Assessment activities	<ul style="list-style-type: none"> Regular low stakes assessment (MCQ, exam questions with peer and self-assessment) Data from low stakes assessment is used to target topics where misconceptions have been identified and to make corrective actions (MRI) Use of online assessment (Forms, Kahoot, Blooket and Quizziz) Content covered is used to form a progress assessment, in the form of an exam style paper with past paper questions from the exam board 		
Resources available	<p>Specification (Page 15 to 18 and 20)</p> <p>OCR GCSE (9-1) Computer Science Specification - J277</p> <p>YouTube (Craig n’ Dave) J277 videos (full specification)</p> <p>GCSE (J277): OCR Specification Order - YouTube</p> <p>BBC Bitesize (Logic Gates)</p> <p>Why data is represented in binary form - Boolean logic - OCR - GCSE Computer Science Revision - OCR - BBC Bitesize</p> <p>KS4 NC information</p> <p>National Curriculum - Computing key stages 3 and 4 (publishing.service.gov.uk)</p>	<p>Specification (Page 14 to 18)</p> <p>OCR GCSE (9-1) Computer Science Specification - J277</p> <p>YouTube (Craig n’ Dave) J277 videos (full specification)</p> <p>GCSE (J277): OCR Specification Order - YouTube</p> <p>BBC Bitesize (Ethical, Legal and Environmental Issues)</p> <p>The impact of computer science technologies - Ethical, legal and environmental impacts of digital technology - OCR - GCSE Computer Science Revision - OCR - BBC Bitesize</p> <p>KS4 NC information</p> <p>National Curriculum - Computing key stages 3 and 4 (publishing.service.gov.uk)</p>	<p>Specification (Page 14 to 18 and 21)</p> <p>OCR GCSE (9-1) Computer Science Specification - J277</p> <p>YouTube (Craig n’ Dave) J277 videos (full specification)</p> <p>GCSE (J277): OCR Specification Order - YouTube</p> <p>BBC Bitesize (Languages and IDEs)</p> <p>High level languages - Programming languages and Integrated Development Environments - OCR - GCSE Computer Science Revision - OCR - BBC Bitesize</p> <p>KS4 NC information</p> <p>National Curriculum - Computing key stages 3 and 4 (publishing.service.gov.uk)</p>
Notes	<p>Strand 1 - This knowledge will build on the foundation for understanding how computers can handle many types of data, but ultimately represent it using just long combinations of 0’s and 1’s. This concept is explored within the basics of logic gates in Year 7 as part of binary and logic, and HT1 of Year 10 where students explore how everything a computer does starts as binary data. Understanding how circuits can be combined will help students to understand how complex data is represented on a computer system, as processed through the FDE cycle (within Year 8 Computing Components and HT3 of Year 10.)</p> <p>Strand 2 - Students will use Boolean logic within programming and build Python representations that can act like logic gates. This is critical to being able to program effectively. Programming skills are constantly developed</p>	<p>Strand 1 – This knowledge is crucial to understanding the impact technology has had, is having and will continue having in the future. It also enables students to recognise the laws that govern technology and the companies that use it. This topic is continually developing and does not contain an exhaustive list of topics it covers, instead it explores a range of ideas and will enable students to tackle additional ideas by considering impacts in categories such as environmental, ethical, cultural and legal issues. This unit also requires students to be able to write reasoned arguments at length, which develops extended writing and structured consideration for advantages and disadvantages.</p> <p>Strand 2 - Students will continue to develop independent programming skills and build solutions in Python to a range of problems. This is critical</p>	<p>Strand 1 – This knowledge links to 1.1, 1.2 and 2.4 in its focus on turning program code (high-level) into something that can be processed by a computer system at its most basic level (low-level machine code). By understanding how high-level is converted into low-level will ensure students understand the basic processing within computer systems and the CPU. The use of IDEs is continual within strand 2, and 2.5 supports students in formally understanding the features that can support programmers in designing efficient programs.</p> <p>Strand 2 - Students will continue to develop independent programming skills and build solutions in Python to a range of problems. This is critical to being able to program effectively. Programming skills are constantly developed and refined throughout all half terms or Year 10 and 11, sometimes linking directly to strand 1.</p>

	and refined throughout all half terms or Year 10 and 11, sometimes linking directly to strand 1.	to being able to program effectively. Programming skills are constantly developed and refined throughout all half terms or Year 10 and 11, sometimes linking directly to strand 1.	
--	--	--	--