Long Term Plan Year 11 OCR J727 Computer Science

Half Term	Unit Title	Key Knowledge/Content to learn and retain	Essential Skills to acquire (subject & generic)	Link to intent and ethos	Anticipated misconceptions	Links to previous KS3	Link to future KS5	Opportunity for stretch and high prior attainers	SMSC & British Values	Cultural Capital	Career Link
One	2.1 Algorithms 2.1.1 Computational thinking 2.1.2 Designing, creating and refining algorithms	Principles of computational thinking: -Abstraction -Decomposition -Algorithmic thinking - Identify the inputs, processes, and outputs for a problem -Structure diagrams Create, interpret, correct, complete, and refine algorithms using: -Pseudocode -Flowcharts -Reference language -Identify common errors -Trace tables	Drawing and labelling abstract diagrams. Technical processes Extended writing Develop a line of enquiry based on observation and provide justification Evaluate benefits and drawbacks Problem Solving Mathematical problems	Learners will develop knowledge and understanding of the practical elements of the course through computational methods directly linked to key knowledge: -Abstraction -Decomposition -Algorithmic thinking Working through a series of learning objectives, learners have the opportunity to improve their resilience through solving problems.	.The best way to learn this unit is by programming • Practice coding small algorithms to help you understand how each programming technique • Make sure loops have a START and END condition • Check > and < signs are the correct way around in loops and IF statements • Many questions in the exam allow you to write your algorithms in short statements or bullet points – you may find this easier	Learners will have developed skills and knowledge from their foundation year of practical programming: -Learners should know the common programming techniques such as: - Subroutines -Data types -Selection -Iteration -Open and close files	Learners will know and be thoroughly prepared for KS5 through practice of solving problems to form a coded solution which leads to an understanding of Object Oriented Programming At A Level and various other data structures.	Learners will have opportunities to learn A Level concepts such as: -Data structures e.g. Queues, lists, hash tables, stacks, graphs through the essential A Level Algorithms guide - Other programming languages such as assembly (low-level), Haskell (Functional), and C# (high-level)	Ethical discussions on open and closed source code and the impacts this has in the industry. Ethical discussions on Ethical hacking by governments and well known hacking groups.	Learners will have the opportunity to discuss the origins of programming: - GCHQ and Bletchley Park -The history of programming from machine code to low level code, to high level code - First computer programmer Ada Lovelace, daughter of Lord Byron the Poet.	https://www.gchq-careers.co.uk/ https://www.mi5.gov.uk/careers https://www.sis.gov.uk/explore-careers.html https://www.yhrocu.org.uk/vacancies/ https://nationalcrimeagency.gov.uk/careers/vacancies
Two	2.1.3 Searching and sorting algorithms	Standard searching algorithms: -Binary Search -Linear search Standard sorting algorithms: -Bubble sort -Merge sort -Insertion sort	Evaluate benefits and drawbacks Compare and contrast algorithms Technical process of each algorithm in terms of its data set	Learners will develop knowledge and understanding of common searching and sorting algorithms through computational methods such as thinking logically and procedurally whilst thinking concurrently when learning how the merge sort works.	Recognise the key words and pseudocode for each algorithm • You may find the OCR Algorithms booklet useful for ideas • Get your peers to make small errors in an algorithm and practise spotting them	Learners will have developed skills and knowledge from their foundation year of practical programming: -Learners should know the common programming techniques such as: - Subroutines -Data types -Selection -Iteration -Open and close files	Learners will know and be thoroughly prepared for KS5 through the knowledge of GCSE common algorithms which leads to an understanding of other Standard algorithms (, insertion sort, , quick sort, Dijkstra's shortest path algorithm, A* algorithm).	Learners will have opportunities to learn A Level concepts such as: -Data structures e.g. Queues, lists, hash tables, stacks, graphs through the essential A Level Algorithms guide	Ethical discussions on open and closed source code and the impacts this has in the industry. Ethical discussions on Ethical hacking by governments and well known hacking groups.	Learners will have the opportunity to discuss the origins of algorithms and programming languages and why so many are still in use today such as Ada, named after Ada Lovelace, which is a programming language used by the military.	GCHQ Degree Apprenticeships Unison Careers
Three	2.2 Programming fundamentals 2.2.2 Data types 2.2.3 Additional programming techniques 2.3.1 Defensive design	The use of variables, constants, operators, inputs, outputs, and assignments -The use of the three basic programming constructs used to control the flow of a program: -Common arithmetic and Boolean operators AND, OR and NOT. -The use of data types -Basic string manipulation -Basic file handling operations -The use of records to store data -The use of SQL to search for data. -The use of arrays -How to use subprograms. Defensive design considerations: -Anticipating misuse -Authentication -Input validation -Maintainability	Drawing and labelling abstract diagrams. Technical processes Extended writing Develop a line of enquiry based on observation and provide recommendations with justification Evaluate benefits and drawbacks	Learners will develop knowledge and understanding of programming fundamentals through computational methods such as thinkinking logically and thinking procedurally when learning about program flow and the use of common mathematical operators, data types, basic string manipulation, and the use of SQL.	Defensive design is about thinking how a program may be misused and how to stop this • Good test plans will test: • That a program works when sensible data is used • How a program works when silly/wrong data is used • That the correct results are given when normal data is used • Some errors stop the program from compiling, like syntax errors • Some errors may only be detected when the program in running: Logic errors, divide by zero	Learners will have developed skills and knowledge from their foundation year of practical programming: -Learners should know the common programming techniques such as: - Subroutines -Data types -Selection -Iteration -Open and close files	This unit underpins the practical elements of programming which leads to a deeper understanding of the theory at KS5 preparing for complex programming techniques such as Object Oriented Programming (OOP). - Learners will be prepared to learn another type of IDE knowing the basics from GCSE e.g. transitioning from IDLE for Python programming language to Visual Studio for multi-platform languages such as C#.	Learners will have opportunities to: -try a different programming language other than Python such as C# using a different IDE. -Create their own problems for their peers to solve. -Complete A Level SQL tasks	Ethical discussions on open and closed source code and the impacts this has in the industry. Ethical discussions on Ethical hacking by governments and well known hacking groups.	Learners will have the opportunity to discuss the origins of algorithms and programming languages and why so many are still in use today such as Ada, named after Ada Lovelace, which is a programming language used by the military	

Half Term	Unit Title	Key Knowledge/Content to learn and retain	Essential Skills to acquire (subject & generic)	Link to intent and ethos	Anticipated misconceptions	Links to previous KS3	Link to future KS5	Opportunity for stretch and high prior attainers	SMSC & British Values	Cultural Capital	Career Link
Four	2.4 Boolean logic 2.5.1 Classification of Languages	Simple logic diagrams using the operators AND, OR, AND NOT -Truth tables -Combining Boolean operators using AND, OR and NOT -Applying logical operators in truth tables to solve problems Characteristics and purpose of different levels of programming language: -High-level languages -Low-level languages -The purpose of translators -The characteristics of a compiler and an interpreter.	Drawing and labelling abstract diagrams. Technical processes Extended writing Develop a line of enquiry based on observation and provide recommendations with justification Evaluate benefits and drawbacks	Learners will develop knowledge and understanding of Boolean algebra and languages through computational methods such as thinking logically and procedurally; - to follow the logic through a given circuit - to follow the timeline of languages	Data is represented in binary due to the 'ON' 'OFF' nature of electricity! These link to pulses of electricity used in computers. • Be sure to remember the difference between MOD and DIV High-level languages look more like English. They are 'further away' from what a computer can understand, e.g. binary (machine code) • Be clear between an assembler, compiler and interpreter. For example, a compiler produces a completed program file. An Interpreter reads the code 'live'. Each one has its use.	Learners will have developed skills and knowledge from their foundation year of some basic understanding of Boolean algebra through the study of practical programming knowing common circuit gates such as AND, OR, and NOT. -Some knowledge of programming language history e.g. the difference between machine code, low-level and high-level languages.	This unit prepares learners for KS5 through the knowledge of common logic gates, they will be able to further their studies on more complex logic gates such as NAND and NOR which leads to an understanding of simplifying Boolean expressions. - Learners will be prepared to complete tasks that require good written communication through comparison of low-level and high-level languages.	- Learners will have opportunities to practise completing truth tables from 3+ inputs - Attempting to draw a logic gate circuit form a given scenario -Attempting to write a Boolean expression from a given complex circuit diagram - Learners are encouraged to complete A Level tasks which includes learning more logic gates.	Learners will have the opportunity to discuss implications of programming languages such as legislation and ethics.	Learners will have the opportunity to discuss the works of Goerge Boole, a self-taught English mathematician, philosopher and logician of whom is credited with laying the foundation of the information age. -The history of programming languages	
Five	2.5.2 The Integrated Development Environment (IDE)	Common tools and facilities available in a integrated Development Environment (IDE): -Editors -Error diagnostics Run-time environment Translators	labelling features of an IDE. Technical processes Extended writing Evaluate benefits and drawbacks	Learners will develop knowledge and understanding of Boolean algebra and languages through computational methods such as thinking logically and procedurally;	There are many different IDEs for stand alone programming languages and ones that are for multiplatform such as Microsoft Visual Studio which can be for Web Development or software development in many different languages.	Once IDLE is mastered at previous key stage, learners may move on to more a sophisticated IDE such as Visual Studio	Learners will be prepared for KS5 Computer Science A Level where learners will have the opportunity to choose a programming language and IDE for their NEA(Non-exam assessment)Programming Project worth 20% towards their overall grade.	Learners will have opportunities to experience different IDEs for different programming languages.	Learners will have the opportunity to discuss implications of programming languages IDEs and ethical implications in Software Development.	-The history of programming languages and IDEs	NHS Software Developer