

HGS Curriculum Map Key Stage 4

Year Group: 10 Computer Science GCSE - Summer

Time period	Summer 1	Summer 2
Topics/ sub-topics	Programming Techniques	Practical Programming Task
Purpose	The unit covers Section 2.2 of the OCR GCSE (9-1) Computer Science specification. The first two lessons focus on data types and arithmetic operations. Followed by an introduction to sequence, selection and iteration. Subsequent lessons discuss arrays, procedures and functions. The final lesson is on records and reading from and writing to a text file.	In this unit students are given the opportunity to undertake an extended programming task. The specification for the programming task will allow them to develop skills within the following areas when programming: <ul style="list-style-type: none"> • Design • Write • Test • Refine
Crucial Learning	<p>At the end of this Unit all students should be able to:</p> <ul style="list-style-type: none"> • 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 Boolean operators in pseudocode solutions • show the results of basic string manipulation functions • 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 • use simple functions and procedures that return values to the calling program <p>Most students will be able to:</p> <ul style="list-style-type: none"> • 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 arrays in the design of solutions to simple problems 	<p>Students will be able to:</p> <ul style="list-style-type: none"> • Explain the programming concepts of sequence, selection and iteration • Understand basic Python syntax and interpret error messages <p>Most students will be able to:</p> <ul style="list-style-type: none"> • Decompose programming problems into smaller parts and then design and apply algorithms to data. • list structures and their various uses. <p>Some students will be able to:</p> <ul style="list-style-type: none"> • To read to and write from text files, and how to structure data so that programs can interpret it correctly.

	<ul style="list-style-type: none"> • write simple functions and procedures using parameters • read from and write to a text file <p>Some students will be able to:</p> <ul style="list-style-type: none"> • explain what is meant by a data structure and why these are used • use two-dimensional arrays in the design of solutions to simple problems • explain why it is good practice to use local variables 	
Sequence	<p>Prior Knowledge</p> <ul style="list-style-type: none"> • In Y7 students learn to program in a visual language e.g. Scratch. In Y9 students complete two programming modules 'Mobile App Development' and an introductory course on 'Python Programming' 	<p>Prior Knowledge</p> <ul style="list-style-type: none"> • In all three of the programming units at KS3, students are required to solve basic programming tasks. They have limited experience of program design or terminal testing.
	<p>Future Learning: A-Level Computer Science (AQA):</p> <ul style="list-style-type: none"> - 4.1 Fundamentals of Programming - 4.2 Fundamentals of Data Structures 	<p>Future Learning: A-Level Computer Science (AQA):</p> <ul style="list-style-type: none"> - 4.13 Systematic Approach to Problem Solving - NEA Computing Practical Project
Skills Acquired	<p>Computational Thinking Skills:</p> <ul style="list-style-type: none"> • Decomposition - breaking down complex problem or system into smaller, more manageable parts • Pattern recognition - looking for similarities among and within problems • Abstraction - focusing on the important information only, ignoring irrelevant detail • Algorithms - developing a step-by-step solution to the problem, or the rules to follow to solve the problem 	<p>Computational Thinking Skills:</p> <ul style="list-style-type: none"> • Decomposition - breaking down complex problem or system into smaller, more manageable parts • Pattern recognition - looking for similarities among and within problems • Abstraction - focusing on the important information only, ignoring irrelevant detail • Algorithms - developing a step-by-step solution to the problem, or the rules to follow to solve the problem

<p>Assessment:</p> <p>Formative & summative</p>	<p>Assessment</p> <p>Verbal Feedback: Regular use of peer, self and teacher feedback</p> <p>Written Feedback: Individual feedback of home learning assessments in the Showbie 'Marking and Feedback' folder. Students have dedicated improvement and reflection time at the start of each lesson.</p> <p>Programming Feedback: The students use an on-line integrated development environment (IDE) called Trinket.io. The software uses versioning to allow students to submit their programming solutions. The teacher has access to a dashboard, that allows them to view, execute and feedback on individual students work.</p> <p>Learning Grids:</p> <ul style="list-style-type: none"> - 2.2 Programming techniques <p>SIMS:</p> <ul style="list-style-type: none"> - SPR 1 OGCU 	<p>Assessment</p> <p>Verbal Feedback: Regular use of peer, self and teacher feedback</p> <p>Written Feedback: Individual feedback of home learning assessments in the Showbie 'Marking and Feedback' folder. Students have dedicated improvement and reflection time at the start of each lesson.</p> <p>Programming Feedback: The students use an on-line integrated development environment (IDE) called Trinket.io. The software uses versioning to allow students to submit their programming solutions. The teacher has access to a dashboard, that allows them to view, execute and feedback on individual students work.</p> <p>Programming Project:</p> <ul style="list-style-type: none"> - Project task sheet provided by OCR <p>SIMS:</p> <ul style="list-style-type: none"> - SPR 2 OGCU
-------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------