



Curriculum Intent

The aim of IT and Computer Science is to develop young people who are digitally literate, resilient and have problem solving skills. Students will develop transferable skills to use across the curriculum, and in the wider world.

$\circ~$ What am I studying this year and how and when will I be assessed?

Торіс	Time of Year	Assessment	Homework
Computational Thinking 2.1 Thinking Abstractly Thinking Ahead Thinking Procedurally Thinking Logically Computational Methods 2.2.2 Features that make a problem solvable by computational methods Problem recognition Problem decomposition Use of abstraction Use of divide and conquer	These pro throughou P They will be structur The prob practice th theoretica and to gain	oblem-solving skills wi at the year in small pro MA lessons and for in e based on the progra res that have been tau lem solving practice a roughout the year wil al knowledge needed n skills to be able to un project in ye	II be taught and practised ogramming projects during dependent study. mming techniques and data ght in lessons with NHA. nd practical programming I help to give context to the to answer exam questions ndertake the programming ear 13.
 Programming Techniques 2.2.1 Use of an IDE to develop/debug programs Programming constructs: Sequence, Selection, Iteration. Global and Local Variables Modularity and its benefits, Functions and Procedures, Parameter Passing by value & by reference. Computational Methods Problem recognition Problem decomposition 	September / October	Initial Assessment Practical programming problems with PMA	Homework will take various forms including the following: exam questions may be set to help develop exam technique; -quizzes may be set to consolidate learning and identify common misconceptions; revision activities (e.g. Isaac Computer Science) may be set to help consolidate learning or revise for assessments; practical work may be set (e.g. coding) to help students develop their practical skills further. A-Level Computer Science students are expected to spend two hours weekly on independent study as well as completing any notes and written or practical homework set.
 Data Structures 1.4.2 Lists and Arrays (1D and 2D) Programming Techniques 2.2.1 Subroutines Files Recursion, its use in comparison to iteration. Computational Thinking 2.1 Thinking abstractly Thinking procedurally / procedural decomposition 	October / November / December	Exam questions	
Revision Computational Thinking 2.1 • Thinking ahead	January January /	Mock exam - paper 2	
 Thinking logically 	rebludiy		

Computational Methods 2.2.2			
 Problem recognition 			
 Problem solving methods 			
Programming techniques for project			
 Working with template files 			
Mini projects			
 Working with files 			
 Websites and templating 			
Algorithms 2.3.1			
 Analysis and design of algorithms 		Exam questions and practical programming project with PMA	
 Standard algorithms (Linear, Binary, 			
Bubble, Insertion, Merge and	March		
Quicksort)			
Measures and methods to determine			
algorithm efficiency, Big O Notation.			
Mini projects			
 Working with files and websites 			
Data Structures 1.4.2			
 1D and 2D Arrays 			
Queues			
 Lists and Linked Lists 		Exam questions and	
 Stacks 	April / May	practical	
 Records, Tuples, and Dictionaries 	April / Iviay	programming project	
Mini projects		with PMA	
 Adding a database 			
 Sample data creation 			
 SQLite tutorial 			
Programming techniques 2.2.1	May / June		
 Object Oriented Programming 	iviay / Julie		
Revision	June	Mock exam - Paper 2	
Programming with GUI's			
 Following tutorials 	June/July		
Č	curre, sury		
Programming Project Preparation			
 Projects introduction 	July		
• What to work on over the summer			
 Project ideas 			

Students will submit work through Google Classroom throughout the year and feedback will be given both verbally and through private comments on Google Classroom to the students where appropriate. Alternatively, feedback will be given in the form of marked exam questions.

End-of-unit assessments will take place upon completion of each unit of study. These will take the form of exam-style questions, sometimes with an additional multiple-choice section.

Revision Guide

Student notes should be used for revision. In order to create these notes, students use the mandatory textbook, as well as Teach-ICT subscription and Isaac Computer Science Access (a variety of sources).

Marking for Literacy	Students will demonstrate pride in their work by:
(Longer answers and written work will be marked for at least one of the below, your teachers will tell you which.)	 Take care of your book and work area. Including Sensible Folder Structure & File Names Title and date all work
Sp – Spelling mistake of key term	 Write as neatly as you can in pen and make sure your electronic
// - Needed new paragraph	files are neat & readable.
C – Capital letter missing	 Update & improve any incorrect work.
P – Punctuation needed	 Upload all work to Google Classroom
Expr - Expression	 Diagrams, graphs, drawings should all be done in pencil or electronically