Curriculum Map: Year 13 Computer Science

Topic	Key Knowledge *What will all students KNOW by the end of the topic?*	Key Skills *What key skills will be learnt/developed by the end of the topic? What will all students be able to DO by the end of the topic?*	Assessment Opportunities *What are the key pieces of assessment? How will students be assessed?*
Elements of computational thinking	**Students will be able to know and understand:** The nature of abstraction The need for abstraction The differences between an abstraction and reality Devise an abstract model for a variety of situations	**Students will:** Apply the criteria across in different contexts including current and future uses of the technologies. Learn advanced programming techniques to use in the creation of independent computer program fir the NEA.	**Students will be assessed by:** Exam questions End of topic / end of section tests
Problem solving and programming	**Students will be able to know and understand:** Programming techniques and programming such as constructs: sequence, iteration, branching Recursion, how it can be used and compares to an iterative approach Global and local variables Modularity, functions and procedures, parameter passing by value and by reference	**Students will:** Apply the criteria across in different contexts including current and future uses of the technologies. Learn advanced programming techniques to use in the creation of independent computer program fir the NEA. Use of an IDE to develop/debug a program	**Students will be assessed by:** Exam questions End of topic / end of section tests

Curriculum Map: Year 13 Computer Science

	Use of object-oriented techniques Computational methods including Problem recognition, Problem decomposition, Use of divide and conquer, Use of abstraction, backtracking, data mining, heuristics, performance modelling, pipelining and visualisation to solve problems		
Algorithms	**Students will be able to know and understand:** Analysis and design of algorithms for a given situation. The suitability of different algorithms for a given task and data set, in terms of execution time and space. Measures and methods to determine the efficiency of different algorithms, Big O notation (constant, linear, polynomial, exponential and logarithmic complexity) Comparison of the complexity of algorithms Algorithms for the main data structures: stacks, queues, trees, linked lists, depth-first (post-order) and breadth-first traversal of trees Standard algorithms: bubble sort, insertion sort merge sort, quick sort Dijkstra’s shortest path algorithm A* algorithm Binary search and linear search	**Students will:** Apply the criteria across in different contexts including current and future uses of the technologies. Learn advanced programming techniques to use in the creation of independent computer program fir the NEA. Create graphs of Big O notation.	**Students will be assessed by:** Exam questions End of topic / end of section tests

Curriculum Map: Year 13 Computer Science

Analysis of the problem	**Students will:** Describe and justify the features that make the problem solvable by computational methods Explain why the problem is amenable to a computational approach Identify and describe those who will have an interest in the solution explaining how the solution is appropriate to their needs (this may be named individuals, groups or persona that describes the target end user) Research the problem and solutions to similar problems to identify and justify suitable approaches to a solution Describe the essential features of a computational solution explaining these choices Explain the limitations of the proposed solution Specify and justify the solution requirements including hardware and software configuration (if appropriate) Identify and justify measurable success criteria for the proposed solution	**Note – NEA project runs linear to the theory aspects of the course**	**Students will be assessed by:** Creation of Analysis section of NEA
Design	**Students will:** Decompose the problem, justifying any decisions made	**Note – NEA project runs linear to the theory aspects of the course**	**Students will be assessed by:** Creation of Design section of NEA

	Explain and justify the structure of the solution Describe the parts of the solution using algorithms justifying how these algorithms form a complete solution to the problem Describe usability features to be included in the solution Identify key variables / data structures / classes justifying choices and any necessary validation Describe the approach to testing Identify the test data to be used during the iterative development and post development phases and justify the choice of this test data		
Developing the Solution	**Students will:** Provide annotated evidence of each stage of the iterative development process justifying any decision made Provide annotated evidence of prototype solutions justifying any decision made Testing to inform development Provide annotated evidence for testing at each stage justifying the reason for the test Provide annotated evidence of any remedial actions taken justifying the decision made	**Note – NEA project runs linear to the theory aspects of the course**	**Students will be assessed by:** Completed NEA

Curriculum Map: Year 13 Computer Science

Evaluation	**Students will:** Provide annotated evidence of testing the solution of robustness at the end of the development process Provide annotated evidence of usability testing (user feedback) Use the test evidence from the development and post development process to evaluate the solution against the success criteria from the analysis Provide annotated evidence of the usability features from the design, commenting on their effectiveness Discuss the maintainability of the solution Discuss potential further development of the solution	**Note – NEA project runs linear to the theory aspects of the course**	
Revision	Revision activities using Isaac computing, course notes, textbook, past papers, Smart Revise etc.		