PCHS Curriculum Information

Course Title: A Level Computer Science	**Exam Board:** OCR	**Specification Code:** H446
How will students be assessed? Unit 1 - 40% - Computer systems written paper - 2h30m - End of Year 13 Unit 2 - 40% - Algorithms and programming written paper - 2h30m - End of Year 13 Unit 3 - 20% - Programming project - 20% - Completed in Year 12 and Year 13		

KEY CONTENT
Half Term 1 **2.1 Thinking abstractly** - The nature of and need for abstraction, The differences between abstraction and reality, Devise an abstract model for a variety of situations, **Thinking ahead** - Identify the inputs and outputs for a given situation, Determine the preconditions for devising a solution to a problem, The need for reusable program components, The nature, benefits and drawbacks of caching, **Thinking procedurally** - Identify the components of and a solution to a problem, Determine the order of the steps needed to solve a problem, Identify sub-procedures necessary to solve a problem, **Thinking logically** - Identify the points in a solution where a decision has to be taken, Determine the logical conditions that affect the outcome of a decision, and how decisions affect flow through a program, **Thinking concurrently** - Determine the parts of a problem that can be tackled at the same time, Outline the benefits and trade-offs that might result from concurrent processing in a particular situation **Dedicated Programming Project Design and Development Lessons**
Half Term 2 **2.2 Programming techniques** - Programming 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 reference, Use of an IDE to develop/debug a program, Use of object-oriented techniques **Dedicated Programming Development Lessons**
Half Term 3 **2.2 Computational methods** - Features that make a problem solvable by computational methods, Problem recognition, Problem decomposition, Use of divide and conquer, Use of abstraction, Learners should apply their knowledge of: backtracking, data mining, heuristics, performance modeling, pipelining, visualisation, ...to solve problems **2.3 Algorithms** - Analysis and design of algorithms for a given situation, Standard algorithms (bubble sort, insertion sort, binary search and linear search), Implement bubble sort, insertion sort, Implement binary and linear search, Representing, adding data to and removing data from queues and stacks, Compare the suitability of different algorithms for a given task and data set, 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) **Dedicated Programming Development Lessons**

Half Term 4 Dedicated Programming Development, Testing, and Evaluation Lessons
Half Term 5 Revision and Exams
Half Term 6