|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Year 12 Term 5**  **A Level Computer Science** | Our mission is to stimulate and challenge our students to excel and provide a desire for lifelong learning and pursue careers in the world of Business, Computing, and ICT. | | | | | |
| **Enquiry Questions: Can you combine different data structures to create a complex one? Could pattern recognition be considered very similar to abstraction?** | | | | | | |
| |  | | --- | | **Component 02: Algorithms & Programming**  This component will incorporate and build on the knowledge and understanding gained in the Computer systems component (01). In addition, learners should understand what is meant by computational thinking, the benefits of applying computational thinking to solving a wide variety of problems, the principles of solving problems by computational methods, be able to use algorithms to describe problems and be able to analyse a problem by identifying its component parts. | | | | | | | |
| **Knowledge**  Students will know about… | **Application/Skills**  Students will be able to… | **Vocabulary** | **Home Learning** | **Assessment** | **Extra Resources**  **Extended Reading** | **Cultural Capital** |
| **1.4.2 Data Structures**  At a more advanced level, static and dynamic data structures are introduced. Together with abstract data types, such as stacks and queues, they allow the more sophisticated organisation and manipulation of data, and provide methods for adding, removing, and traversing data.  **2.1 Elements of computational thinking Understand what is meant by computational thinking**  Learning how to solve problems in various computational methods. Breaking down larger problems into smaller solvable tasks. | * distinguish between an array, list and tuple * describe the addition, deletion and maintenance of data within queues, stacks, hash tables and trees * describe the characteristics of an array-based queue, circular queue and priority queue * write an algorithm for traversing a linked list * be able to compare the use of adjacency matrices and adjacency lists for representing graphs * be able to apply a number of different hashing algorithms * describe the nature of and need for abstraction * devise an abstract model for a variety of situations * design algorithms to solve complex problems * hand trace a complex algorithm to say what it does * determine the parts of a problem that can be executed concurrently * outline the benefits and trade-offs that might result from concurrent processing in a particular situation * apply techniques of backtracking, data mining, heuristics, performance modelling, pipelining and visualisation to the solution of problems | * Abstraction * Decomposition * Pattern Recognition * Procedural Programming * Logical Programming * Functional Programming * Linked List * Array * Tuple * Dictionary * 2/3 dimensional array * Array iterations * Array operations * Queue * Stack * LIFO queue * FIFO queue * Undirected graph * Directed graph * Weighted graph * Adjacency Matrix * Adjacency List * Breadth First Search * Depth First Search * Trees * Tree traversal * Hash tables * Hash Functions * Load Factor * Search algorithm * Linear Probe * Chaining | High quality Homework set on Google Classrooms  Teach-ICT.com  PG Online – ClearRevise A Level Computer Science | End of unit assessments via Socrative  In-class mock exams | [Teach-ICT.com](https://teach-ict.com/2016/GCSE_Computing/OCR_J277/OCR_J277_home.html)  [Isaac Computer Science](https://isaaccomputerscience.org/topics/gcse?examBoard=all&stage=all#all)  Seneca – [Computer Science](https://app.senecalearning.com/classroom/course/a1ce4570-6e27-11e8-af4b-35cf52f905c2/section/65ac2e24-3b57-4598-b4dc-01e04eddee1b/session)  BBC Bitesize  Hodder Education – Revision Book A Level Computer Science | The National Science Museum (free events)  <https://www.sciencemuseum.org.uk/>  The Royal Institute of Science (free events)  <https://www.rigb.org/families/family-fun-days>  **National Museum of Computing, Bletchley Park (Near Milton Keynes)**  <http://www.tnmoc.org/>    <https://www.bletchleypark.org.uk/>  <http://www.codesandciphers.org.uk/bletchleypark/>  (virtual tour)    Centre for Computing History, Cambridge  <http://www.computinghistory.org.uk/> |