|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Year 12 Term 6**  **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: Is Dikjtra’s Algorithm useful when all nodes are of equal length?** | | | | | | |
| |  | | --- | | **Component 02: Algorithms & Programming**  This component will introduce learners to the internal workings of the Central Processing Unit (CPU), the exchange of data and will also look at software development, data types and legal and ethical issues. It is expected that learners will draw on this underpinning content when studying computational thinking, developing programming techniques and devising their own programming approach in the Programming project component (03 or 04). Learners will be expected to apply the criteria below in different contexts including current and future uses of the technologies. | | | | | | | |
| **Knowledge**  Students will know about… | **Application/Skills**  Students will be able to… | **Vocabulary** | **Home Learning** | **Assessment** | **Extra Resources**  **Extended Reading** | **Cultural Capital** |
| **2.2 Problem solving and programming**  **2.2.1 Programming techniques**  This unit covers the use of an IDE to develop/debug a program, and the fundamentals of programming, while recognising that some students may have had little previous experience of programming and others will already be seasoned programmers  **2.3 Algorithms**  This is a theoretical unit covering Algorithms (except algorithms for stacks, queues, trees and linked lists). Searching and sorting algorithms (bubble sort, insertion sort, merge sort, quick sort) are explained in an interactive and practical way, with reference to Big-O notation in terms of time and space complexity. Further topics tackles standard algorithms for depth-first and breadth-first graph traversals. Optimisation algorithms, such as Dijkstra’s shortest path algorithm and the A\* algorithm are covered along with a discussion of intractable problems, in the final topic.  **3.1. Programming Project - Analysis of the problem**  Students at this stage will start to think about their project and identify a problem which needs solving. Students will need to write up a detailed analysis of their problem and discuss their solution. | • describe features of an IDE which are useful in developing and debugging a program  • write a pseudocode solution for a problem involving iteration and selection (branching)  • determine the output from a pseudocode program  • use structured programming techniques and write their own subroutines with parameters  • construct algorithms using two-dimensional arrays  • use local and global variables in subroutines  • trace through a recursive algorithm  • compare iterative and recursive algorithms for solving a problem  • complete given pseudocode for an object oriented program  • state the time complexity of an algorithm  • write an algorithm for a linear search  • trace through a bubble sort algorithm  • trace through an insertion sort algorithm  • explain how the merge sort works and analyse its time complexity  • explain how the quicksort works  • describe applications of each graph traversal  • be able to trace Dijkstra’s shortest path algorithm  • Give examples of intractable problems  • Describe briefly the A\* algorithm and its purpose | * Sequence * Selection * Iteration * Variables * Syntax * Arrays * Libraries * Space efficiency * Time efficiency * Compromises * Analysing algorithms * Big O as functions * Complexity * Simplifying * Inner loops * Main term * Comparing algorithms * Linear Sequential search * Binary search * Bubble sort * Insertion sort * Merge sort * Quicksort * Dikjstra's algorithm * A\* algorithm | 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/> |