|  |  |
| --- | --- |
| **YEAR 12 A LEVEL COMPUTER SCIENCE SUMMER TERM 3 – PAPER 2**  **‘An ambitious curriculum that meets the needs of all’**  **Medium Term Planning – Data Structures** | |
| **Curriculum Intent** | **Pupils will be taught the following National Curriculum guidelines this term:**  **At the end of this Unit all students should be able to:**   * describe the concept and uses of a * Queue * Stack * Graph * Graph Depth First and Breath First * Tree pre, post, in order * Binary search tree * Hash table * list typical uses of each of these data structures * know how an adjacency matrix and an adjacency list may be used to represent a graph * traverse a binary tree in pre-order, in-order and post-order * create a binary search tree * be able to apply a simple hashing algorithm * state what is meant by a collision and describe how collisions may be handled * state a possible order in which nodes are visited in depth first and breadth first graph traversals   **Most students will be able to:**   * 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   **Some students will be able to:**   * describe and apply the following operations to a linear, circular and priority queue: * add an item * remove an item * test for empty queue * test for full queue * write algorithms for adding and deleting elements to/from a linked list * write algorithms for pre-order, in-order and post-order tree traversals * trace depth-first and breadth-first graph traversal algorithms |
| **Skills/Assessment Objective Links** |
| **Numeracy** | MOD, DIV, <, > |
| **Literacy** | **Vocabulary Tier 3:** Elementary data type, composite data type, abstract data type, static data structure, dynamic data structure, heap, overflow, underflow, array, record, tuple, list, linked list, queue, circular queue, priority queue, First In, First Out (FIFO), enqueue, dequeue, append, push, pop, stack, Last In, First Out (LIFO), call stack, stack frame, parameter, return address, hashing, hash table, collision, mid-square method, folding method, dictionary, graph, edge, arc, vertex, node, directed graph, digraph, undirected graph, weighted edge, adjacency matrix, adjacency list, Page Rank algorithm, tree, root, child, parent, subtree, leaf node, binary search tree, pre-order, in-order and post-order traversal, depth-first traversal, breadth-first traversal, pre-order and post-order tree traversal optimisation problem  **Vocabulary Tier 2:** weight, edge, list, pre order, post order, problem  **Reading:**  Worksheets, presentations, answer sheets, exam questions, mark scheme, further reading for homework, conduct research for NEA  **Writing**: Answer on the worksheet via word, complete NEA  **Oracy:** listening and using tier 3 words |
| **Becoming future ready** | **Careers/Employability:**  Understand the grade requirements at universities and the topics that can be applied for. Explore apprenticeship opportunities with a range of industries.   * Software Architect. * Data Scientist. * Machine Learning Engineer. * Blockchain Developer * Cybersecurity Engineer. * Cloud Solutions Architect. * AI Research Scientist. * Full-Stack Developer. |
| **Adaptation** | Throughout this topic, quality first teaching will provide differentiation:  **By product:** Learners are asked to present outcomes writing different code, not all code will be equal in style and sophistication, all code will work with teachers input, top end programmers will be set challenges on how to extend code and be expected to conduct a level of independent research. Learners are asked to present outcomes in a different way via pieces of writing, targeted questioning, models and drawings and speaking.  **By resource:** Worksheets are well presented and accessible. Instructions are clearly outlined and separate from the information so that pupils know where to begin and end. Handouts are differentiated by outcome. Resources used will appeal to the range of preferred learning styles of pupils e.g. visual, auditory or kinesthetic learners. Scaffolding of tasks – word frames.  **By Intervention:** By providing different levels of supervision and support  **By Progressive Questioning:** Exploring pupils’ understanding through interactive dialogue using Blooms Taxonomy.  **By Grouping:** According to prior coding attainment, gender, social preference, preferred learning style.  **By Task:** Pupils identify targets which are meaningful via level of coding ability and feedback sheets  **By Offering Optional Activities**: In class or as homework, to extend learning.  This QFT/SEND provision will be explicit within the lesson by lesson schemes of work. |
| **QFT/SEND Provision** |
| **Implementation**  **Curriculum Delivery** | To be able to:  **Topic 1 Queues FIFO**  Understand the concept of an abstract data type  Be familiar with the concept and uses of a queue  Describe the creation and maintenance of data within a queue (linear, circular, priority)  Describe and apply the following to a linear, circular and priority queue  o Add an item  o Remove an item  o Test for an empty queue  o Test for a full queue  **Topic 2 Stacks LIFO**  Be familiar with the concept and uses of a stack  Be able to describe the creation and maintenance of data within a stack  Be able to describe and apply the following operations:   * Push * Pop * Peek (or top) * Test for empty stack * Test for full stack   Be able to explain how a stack frame is used with subroutine calls to store return addresses, parameters and local variables  **Topic 3 Trees**  Know that a tree is a connected, undirected graph with no cycles  Know that a binary tree is a rooted tree in which each node has at most two children  Be familiar with typical uses for rooted trees  **Topic 4 Graphs**  Be aware of a graph as a data structure used to represent complex relationships  Be familiar with typical uses for graphs  Be able to explain the terms: graph, weighted graph, vertex/node, edge/arc, undirected graph, directed graph  Know how an adjacency matrix and an adjacency list may be used to represent a graph  Be able to compare the use of adjacency matrices and adjacency lists  **Topic 5 Graphs Traversal Algorithm - Depth First**  Be able to trace depth-first algorithms  Be able to trace breadth-first algorithms  Describe typical applications of each  **Topic 6 List and Linked Lists**  Explain how a list may be implemented as a static or dynamic data structure  Describe the linked list data structure  Show how to   * Create * Traverse * Add data to * Remove data from a linked list   **Topic 7 Hash Tables**  Be familiar with a hash table and its uses  Be able to apply simple hashing algorithms  Know what is meant by a collision and how collisions are handled using rehashing  Be familiar with the concept of a dictionary  Be familiar with simple applications of a dictionary  End of unit assessment |
| **Learning Outcomes (Knowledge)** |
| **Current learning to be developed in the future within:** | Links into understanding algorithms and bring in programming techniques |
| **Assessment** | See assessment maps for formative and summative assessment opportunities. |
| **Impact** | Review assessment results and target pupils that require further support via:-   * Learning conversation * Changing seating plan * Plan lessons to address areas of concern in assessment * Targeted homework based on low performance areas identified in the assessment and marked pieces * Stretch and challenge high ability pupils by identifying ambitious next steps to expand knowledge   Create a feedback sheet for each student  Each student identifies areas of Green, Amber and Red using Mark Assessment on their feedback sheet  Complete NOW task on areas identified as Amber and Red |

