**Curriculum Map: Computer Science Year 11**

| | **Autumn** | **Spring /Summer** |
|---|---|---|
| **Content**<br>Declarative knowledge<br>‘I Know’ | 2d. Practical Programming Skills<br>2.1.1 Computational thinking<br>2.1.2 Designing, creating and refining algorithms (subtopics: structure diagram, ERL consolidation)<br>2.1.3 Searching and sorting algorithms<br>2.2.3 Additional programming techniques (subtopic: SQL)<br>2.3.1 Defensive design<br>2.4.1 Boolean logic<br>2.5.1 Languages<br>2.5.2 The Integrated Development Environment (IDE) | Revision of all topics<br>Exam strategy<br>Practising Past Papers |
| **Skills**<br>Procedural Knowledge<br>‘I know how to’ | ☐ Phases of the Software Development Lifecycle<br>o Analysis<br>o Design<br>o Development<br>o Testing<br>o Evaluate<br>o Refine<br>☐ Principles of computational thinking:<br>○ Abstraction<br>○ Decomposition<br>○ Algorithmic thinking<br>☐ Structure diagrams<br>☐ ERL consolidation<br>☐ Standard searching algorithms:<br>○ Binary search<br>○ Linear search<br>☐ Standard sorting algorithms:<br>○ Bubble sort<br>○ Merge sort<br>○ Insertion sort<br>☐ The use of SQL to search for data<br>☐ Defensive design considerations:<br>○ Anticipating misuse<br>○ Authentication<br>☐ Input validation<br>☐ Maintainability:<br>○ Use of sub programs<br>○ Naming conventions<br>○ Indentation<br>○ Commenting<br>☐ Simple logic diagrams using the operators AND, OR and NOT<br>☐ Truth tables<br>☐ Combining Boolean operators using AND, OR and NOT<br>☐ Applying logical operators in truth tables to solve problems<br>☐ Characteristics and purpose of different levels of programming language:<br>○ High-level languages | • Use Mark Schemes to mark answers and improve them<br>• Use exam techniques applicable to the subject, for example,<br>○ identifying the hierarchy of command verbs like Name, Define, Describe, Compare, Justify etc.<br>○ identify the exact topic and sub-topics the question refers to.<br>○ identifying the context to apply to the question<br>○ how many marks the question is, 1 unique point to state usually per mark<br>○ Practise questions under timed conditions |

| | | |
|---|---|---|
| | ○ Low-level languages<br>□ The purpose of translators<br>□ The characteristics of a compiler and an interpreter<br>□ Common tools and facilities available in an Integrated Development Environment (IDE):<br>○ Editors<br>○ Error diagnostics<br>○ Run-time environment<br>○ Translators | |
| **Strategies**<br>Conditional<br>Knowledge<br>‘I know when to’ | • Be able to solve a big problem by applying all the principles of computational thinking, suggesting which techniques are most appropriate for certain problems<br>• Suggest validation techniques a program should use based on different requirements<br>• How to create, complete or edit logic diagrams, including multiple gates, and truth tables for given scenarios<br>• Able to identify levels of languages, and suggest the suitability of a language or not for running on different systems architectures | |
| Key Questions | ✓ Understanding of these principles and how they are used to define and refine problems<br>✓ Understand the main steps of each algorithm<br>✓ Understand any pre-requisites of an algorithm<br>✓ Apply the algorithm to a data set<br>✓ Identify an algorithm if given the code for it<br>✓ SQL commands:<br>▪ SELECT<br>▪ FROM<br>▪ WHERE<br>✓ Understanding of the issues a programmer should consider to ensure that a program caters for all likely input values<br>✓ Understanding of how to deal with invalid data in a program<br>✓ Authentication to confirm the identity of a user<br>✓ Practical experience of designing input validation and simple authentication (e.g. username and password)<br>✓ Understand why commenting is useful and apply this appropriately<br>✓ Knowledge of the truth tables for each logic gate<br>✓ Recognition of each gate symbol<br>✓ Able to create, complete or edit logic diagrams and truth tables<br>✓ Ability to work with more than one gate in a logic diagram<br>✓ The differences between high- and low-level programming languages<br>✓ The need for translators<br>✓ The differences, benefits and drawbacks of using a compiler or an interpreter<br>✓ Knowledge of the tools that an IDE provides<br>✓ How each of the tools and facilities listed can be used to help a programmer develop a program<br>✓ Practical experience of using a range of these tools within at least one IDE | |
| Assessment topics | September assessment (baseline)<br>Nov-Dec PPE in both CS papers | Mar-Apr PPE in both papers |

| | | |
|---|---|---|
| Cross curricular links/Character Education | Ability to work in teams. Logical Thinking, Resilience, Problem solving. | |
| | | |