# Computer Science: Curriculum for Sept 2021

## Key Stage 4: Year 9  2021/2022

| Term | Topic | Covered in lessons | Intent | NC Focus 1 | NC Focus 2 | Assessment |
|---|---|---|---|---|---|---|
| HT 1 | **2.1 Algorithms** <br><br> **Computational Thinking** | • Decomposition <br> • Abstraction <br> • Pattern Recognition <br> • Algorithms | Computational Thinking forms the foundation for the entire course. Embedding these skills will allow students to be able to approach real world problems logically and understand the workings of the computer | Design, use and evaluate computational abstractions that model the state and behaviour of real-world problems and physical systems | Understand several key algorithms that reflect computational thinking [for example, ones for sorting and searching]; use logical reasoning to compare the utility of alternative algorithms for the same problem | Project Based: <br> Solving a Systems Architecture problem logically using the <br><br> stages of Computational Thinking |
| | **1.1 Systems Architecture:** <br><br> **1.1.1 Architecture** <br><br> **1.1.2 – CPU Performance** <br><br> **1.1.3 - Embedded systems** | • Systems Architecture <br> • Purpose of the CPU <br> • Von Neuman <br> • Components/characteristic <br> • FDE <br> • RAM/ROM | Understand the terms and processes in computational thinking and be able to use the skills of abstraction, decomposition and algorithmic thinking. | understand the hardware and software components that make up computer systems, and how they communicate with one another and with other systems | | Project Based: |
| HT2 | **2.2 Programming :** <br><br> **Intro to Programming** <br><br> **- Physical (Microbits)** | • Variables <br> • Lists <br> • Selection <br> • Iteration-FOR and WHILE Loops <br> • Algorithms <br> • Designing, <br> • Creating and refining algorithms <br> - Flowcharts <br> - Pseudocode | Students develop, apply and practice, analytical, problem-solving, design, and computational thinking skill with hands on practical computing devices <br><br> Further develop flowcharts and pseudocode | Use two or more programming languages, at least one of which is textual, to solve a variety of computational problems; make appropriate use of data structures [for example, lists, tables or arrays]; design and develop modular programs that use procedures or functions | Develop and apply their analytic, problem-solving, design, and computational thinking skills | Project Based: |

# Computer Science: Curriculum for Sept 2021

## Key Stage 4: Year 9  2021/2022

| Term | Topic | Covered in lessons | Intent | NC Focus 1 | NC Focus 2 | Assessment |
|---|---|---|---|---|---|---|
| HT 3 | **1.2 Memory and Storage:**<br><br>**1.2.1 – primary memory** | • Primary storage<br>• RAM and ROM<br>• Virtual memory | Learn where different types of data can be stored | understand the hardware and software components that make up computer systems, and how they communicate with one another and with other systems<br>understand how instructions are stored and executed within a computer system; | understand how instructions are stored and executed within a computer system; understand how data of various types (including text, sounds and pictures) can be represented and manipulated digitally, in the form of binary digits | End of Topic Test and Exam questions |
| | **1.2.2 - Secondary memory** | • Types of Storage<br>• Characteristics of storage | Learn about external storage | understand the hardware and software components that make up computer systems, and how they communicate with one another and with other systems<br>understand how instructions are stored and executed within a computer system; | understand how instructions are stored and executed within a computer system; understand how data of various types (including text, sounds and pictures) can be represented and manipulated digitally, in the form of binary digits | End of Topic Test and Exam questions |
| | **1.2.3 Units**<br><br>**1.2.4 – Data Storage**<br><br>**1.2.5 - Compression** | Data Representation<br>• Units of data<br>• Data storage<br>• Character sets<br>• Images<br>• (Sound)<br>• Compression | Learn how computers understand and make use of data | understand how instructions are stored and executed within a computer system; understand how data of various types (including text, sounds and pictures) can be represented and manipulated digitally, in the form of binary digits | | End of Topic test and Exam questions |

# Computer Science: Curriculum for Sept 2021

## Key Stage 4: Year 9  2021/2022

| Term | Topic | Covered in lessons | Intent | NC Focus 1 | NC Focus 2 | Assessment |
|---|---|---|---|---|---|---|
| HT4 | **2.5 Programming languages and integrated development Environments**<br><br>**Languages(Translators and Facilitators) IDE, SQL** | High / Low level Low<br>•Translators<br>•Compiler / Interpreter | Practical use of the techniques in a high-level language<br><br>Practical use of the data types in a high-level language<br><br>Practical use of the additional programming techniques I<br><br>Develop the fundamental techniques and concepts of text-based programming. | Use two or more programming languages, at least one of which is textual, to solve a variety of computational problems; make appropriate use of data structures [for example, lists, tables or arrays]; design and develop modular programs that use procedures or functions | Develop and apply their analytic, problem-solving, design, and computational thinking skills<br><br>understand several key algorithms that reflect computational thinking [for example, ones for sorting and searching]; use logical reasoning to compare the utility of alternative algorithms for the same problem | Project Based<br><br>Flowcharts<br>Pseudocode<br>Algorithms<br>Programming<br><br>End of Topic test:<br>High/Low level languages |
| | **Text Based Programming**<br><br>**2.1.2 Designing, creating and refining algorithms**<br><br>**2.2 Programming**<br><br>**2.2.1 Fundamentals**<br><br>**2.2.2 Data types** | • Pseudocode<br>• Flowcharts<br>• Reference language/high-level programming language<br><br>• The use of variables, constants, operators, inputs, outputs and assignments ¨<br><br>• Basic programming constructs:<br>  - Sequence<br>  -Selection<br>  - Iteration<br><br>• Boolean operators AND, OR and NOT | Also, the opportunity to link the physical programming principles and techniques learnt into text-based programming.<br><br>Develop Flowcharts and Pseudo coding skills and techniques | | | |

# Computer Science: Curriculum for Sept 2021

## Key Stage 4: Year 9  2021/2022

| Term | Topic | Covered in lessons | Intent | NC Focus 1 | NC Focus 2 | Assessment |
|------|-------|-------------------|--------|-----------|-----------|-----------|
| HT5 | **2.4 Boolean Logic** | • AND/OR/NOT Gates<br><br>• Truth tables | Understand why data needs to be in binary form and how transistors in computers are used to make decision | computational thinking [for example, ones for sorting and searching]; use logical reasoning to compare the utility of alternative algorithms for the same problem] | Understand simple Boolean logic [for example, AND, OR and NOT] and some of its uses in circuits and programming; understand how numbers can be represented in binary, and be able to carry out simple operations on binary numbers [for example, binary addition, and conversion between binary and decimal | |
| HT6 | **Programming Project**<br><br>**2.2.1 Programming fundamentals**<br><br>**2.2.3 Additional programming Techniques**<br><br>**2.3 Producing robust Programs**<br><br>**2.3.1 Defensive Design**<br><br>**2.3.2 Testing** | Programming project with Flow and Pseudocode<br><br>A programming scenario is shared with students, and they are asked to develop a solution to that through the following:<br><br>• Analysis of the problem<br>• Design a solution<br>• Programming Techniques – showcase a range of techniques suitable to the problem.<br>• Development – Show how the program comes together.<br>• Evaluation and Testing – Evaluate the effectiveness of the program and how it meets the given problem. Fully test all elements of the program. | Students learn the fundamentals of programming in Python | use two or more programming languages, at least one of which is textual, to solve a variety of computational problems; make appropriate use of data structures [for example, lists, tables or arrays]; design and develop modular programs that use procedures or functions | develop and apply analytic, problem-solving, design, and computational thinking skills | Practice exam questions |