Computer Science CURRICULUM OVERVIEW

Key Stages 3 & 4

YEAR	TOPICS	TERMLY INDEPENDENT WORK
7	1. Autumn Term 1.1. Basic Digital Skills 1.2. Components of a Computer System 1.3. Data Representation Techniques 2. Spring Term 2.1. Logic Gates 2.2. Numerical Modelling 3. Summer Term 3.1. Computational Thinking 3.2. Introduction to Python	1. Autumn Term 1.1. Research into the history of Computer Science 1.2. Designing a computer system challenge 1.3. Investigation into the impact of encryption 2. Spring Term 1.1. Investigate Redstone Logic in Minecraft 2.1. Modelling exercise: Party planner 2.2. Python: Chilli Challenges (Turtle) 3. Summer Term 3.1. Python: Chilli Challenges (Quizzes)
8	1. Autumn Term 1.1. Computer Architecture and Assembly Language 1.2. Digital Image Manipulation 1.3. Legislation in a digital world 2. Spring Term 2.1. Flowcharts and Control 2.2. Minecraft modification with Python 3. Summer Term 3.1. Abstract Data Structures 3.2. Basic Searching and Sorting Techniques	1. Autumn Term 1.1. LMC: Chilli Challenges 1.2. Digital Art Challenge 1.3. Investigation into embedded systems 2. Spring Term 2.1. Minecraft 'Mini-game' and 'House of Code' competitions 2.2. Comparing Operating Systems 2.3. Python: Chilli Challenges (Loops) 3. Summer Term 3.1. BubbleSort Challenge

9	1. Autumn Term 1.1. Networking – Topologies and Protocols 1.2. Databases: Theory and SQL 2. Spring Term 2.1. Modular programming in Python 2.2. Introduction to Object Oriented programming 3. Summer Term 3.1. Physical Computing 3.2. Game design project 3.3.	1. Autumn Term 1.1. Develop revision guide on networking 1.2. SQL investigation 2. Spring Term 2.1. Building adding circuits 2.2. PgGame and Tkinter challenges 3. Summer Term 3.1. Robotics investigations 3.2. Open Ended programming project
10	1. Computational Thinking 2. Algorithm design 3. Programming Techniques 4. Data Representation 5. Systems Architecture 6. Memory and Storage 7. Wired and Wireless networks 8. Topologies, protocols and layers	1) A wide range of extension programming tasks are available throughout the year, requiring students to develop their analytical, design and development skills within Python 2) Develop revision materials in collaborative work area 3) Investigation into Assembly Language 4) Investigation of alternative languages
11	1. Programming project 2. System software 3. System Security 4. Computational logic 5. Translators and facilities of languages	1) Develop revision materials in collaborative work area 2) Research directly relating to the programming project

PLEASE NOTE:

- This overview sets out a general summary of the basic curriculum taught. It is not an exhaustive list of what may be taught and subject teachers may follow the above in a different order. Further details may be obtained from the Head of Department, if required.
- The Independent Work indicated represents core, headline tasks per term; weekly/fortnightly independent/home work is set in all subject areas, and details are noted in Pupil Planners.