Key Stage 4 LTP

|  |
| --- |
| **Curriculum Intent** |
| The course has a written and a practical element. This balance provides students with a rounded experience of computer science in comparison with a traditional paper-based exam that relies on students reading and writing pseudocode. The new onscreen assessment will give them a more practical and engaging assessment experience that can be transferred into real world skills.   * A practical onscreen assessment approach will prepare students well for eventual employment * follows the practical teaching and learning approach of the interactive scheme of work * It gives an opportunity for the learner to be assessed on writing, testing, and refining programs in an IDE using Python 3. * Being given time constraints to complete a task also relates well to the real-life time constraints experienced by many Computer Scientists.   **Paper 1 Principles of Computer Science Paper code: 1CP2/01 75 marks Written examination: 1 hour 30 minutes 50% of the qualification.**   * Students will be assessed on 5 Topics:  1. **Computational thinking** - understanding of what algorithms are, what they are used for and how they work; ability to follow, amend and write algorithms; ability to construct truth tables.  2. **Data** - understanding of binary, data representation, data storage and compression.  3. **Computers** - understanding of hardware and software components of computer systems and characteristics of programming languages.  4. **Networks** - understanding of computer networks and network security.  I 5. **Issues and impact** - awareness of emerging trends in computing technologies, and the impact of computing on individuals, society and the environment, including ethical, legal and ownership issues.   **Paper 2 Application of Computational Thinking Paper code: 1CP2/02 75 marks Practical onscreen examination: 2 hours 50% of the qualification.**   * Students will be assessed on Topic 6: Problem solving with programming. The main focus of this paper is:  1. Understanding what algorithms are, what they are used for and how they work in relation to creating programs 2. Understanding how to decompose and analyse problems 3. Ability to read, write, refine and evaluate programs. |

|  |
| --- |
| **Prior Learning Summary** |
| * Year 7: Completion of the topics covered: Understanding Computers, Staying Safe Online, An Introduction to Algorithms * Year 8: Completion of the topics covered: Understanding the Internet, An Introduction to Binary, An introduction to Coding |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Curriculum Structure – Year 9** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | A | A | A | A | A | A | A | A | A | A | A | A | B | B | B | B | B | B | E | E | E | E | E | E |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Curriculum Structure – Year 10** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| B | B | B | B | C | C | C | C | C | C | C | C | C | D | D | D | D | D | D | D | D | D | D | D | E | E | E | E | E | E | E | E | E |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Curriculum Structure – Year 11** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| E | E | E | E | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F |  | F |  |  |  |  |  |  |  |  |  |  |  |

|  |  |
| --- | --- |
| **Unit A – Computational Thinking (Problem Solving)** | |
| **Overview** | * **1.1 Decomposition and Abstraction:** * 1.1.2 Benefits of subprograms * **1.2 Algorithms:** * 1.2.2 Variables and Data Structures * 1.2.3 Operators * 1.2.4 Trace Tables * 1.2.5 Errors * 1.2.6 Sort and Search * 1.2.7 Efficiency of Algorithms * **1.3 Truth Tables** |
| **Aims** | * Understanding of what algorithms are, * Understand what algorithms are used for and how they work * Able to follow, amend and write algorithms; * Able to construct truth tables. |

|  |  |
| --- | --- |
| **Unit B – Data** | |
| **Overview** | * **2.1 Binary** * 2.1.1 Needs for Binary * 2.1.2 Representing Integers * 2.1.3 Denary to Binary * 2.1.4 Binary Arithmetic * 2.1.5 Overflow * 2.1.6 Hexadecimal * 2.2 Data Representation * 2.2.1 ASCII * 2.2.2 Bitmap Representation * 2.2.3 Sound Representation * 2.2.4 Bit Constraints * 2.3 Data Storage and Compression * 2.3.1 Units of Storage * 2.3.2 Compression |
| **Aims** | * Understanding of binary, Denary, Hexidecimal (base2, base 10, base16) * Understand data representation (understanding the different forms of; text, images and sound) * Understanding data storage (formats, Specific components, dealing with memory recording) * Understanding the two terms of compression (Lossy and Lossless). |

|  |  |
| --- | --- |
| **Unit C – Computers** | |
| **Overview** | 3.1 Hardware  3.1.1 Systems Architecture  3.1.2 Secondary Storage  3.1.3 Embedded Systems  3.2 Software  3.2.1 Operating System  3.2.2 Utility Software  3.2.3 Robust Software  3.3 Programming Languages  3.3.1 Levels of Language  3.3.2 Translators |
| **Aims** | * Understanding of hardware and software * Continuation from Unit B ‘Data’, general components of computer systems * Know the characteristics of programming languages (High Level, Low Level, Machine Level) |

|  |  |
| --- | --- |
| **Unit D – Networks** | |
| **Overview** | **4.1 Networks**  4.1.1 Needs for Networks  4.1.2 Types of Networks  4.1.3 Internet Structure  4.1.4 Wired and Wireless networks  4.1.5 Network Performance  4.1.6 Protocols  4.1.7 TCP/IP  4.1.8 Network Topologies  **4.2 Network Security** |
| **Aims** | * Have an understanding of computer networks (types of topologies and layout) * Understand the threats to network security and have and know the measures required to help secure networks from threats. |

|  |  |
| --- | --- |
| **Unit E – Issues and Impact** | |
| **Overview** | 5.1 Environmental  5.1.1 Environmental  5.2 Ethical and Legal  5.2.1 Personal Data  5.2.2 Artificial Intelligence  5.2.3 Intellectual Property  5.3 Cyber Security  5.3.1 Security Threats  5.3.2 Securing Systems |
| **Aims** | * Have an awareness of emerging trends in computing technologies, * Have an understanding of the impact of computing on individuals, society and the environment * Have an awareness of the ethical, legal and ownership issues surrounding technology, data and privacy. |

|  |  |
| --- | --- |
| **Unit F – Problem Solving with Programming** | |
| **Overview** | 6.1 Develop Code  6.1.1 Decomposition and abstraction  6.1.2 Write Programs  6.1.3 Convert algorithms and programs  6.1.4 Maintainable Code  6.1.5 Find Program Errors  6.1.6 Evaluate Program Efficiency  6.2 Constructs  6.2.1 Understanding program structures  6.2.2 Sequence, selection, repetition  6.3 Data Types and Structures  6.3.1 Data Types and Structures  6.3.2 Variables and constants  6.3.3 Strings  6.4 Input/Output  6.4.1 User Input  6.4.2 Text Files  6.4.3 Validation  6.4.4 Authentication  6.5 Operators |
| **Aims** | * Understanding what algorithms are, what they are used for and how they work in relation to creating programs * Understanding how to decompose and analyse problems * Ability to read, write, refine, and evaluate programs. |