Curriculum Intent Statement for Computer Science

At Chase Terrace Academy we aspire for all of our students to achieve greater things than they ever thought possible.

We pride ourselves on being a warm and welcoming school that places community at the heart of everything we do. Our ambitious curriculum is enriching and inclusive, providing challenge and breadth for all. This empowers our students to become compassionate, confident and creative individuals who are resilient, respectful and equipped with a desire to take up a fulfilling role in society and the wider world.

In Computer Science we aspire to enrich students with a varied and deep understanding of computing developments, concepts and the impact of technology on our society and environment. Students learn a diverse range of skills such as programming in a range of languages and also study the theory behind the science of computing, the Internet and the ever growing importance of our personal security and privacy. Ultimately, we aim to give students the knowledge and experience they need to study Computing to degree level, to use technology in their day to day lives or careers and to manipulate technology and tools to compliment almost any future study or job.

Year 10 Curriculum Implementation Plan (Computer Science)

Computer Science				
Knowledge and Skills – Students will be have studied…	Reading, Literacy and Numeracy	Formative Assessment	Summative Assessment	Link to GCSE Content
• The purpose of the CPU • Von Neumann architecture: ◯ MAR (Memory Address Register) ◯ MDR (Memory Data Register) ◯ Program Counter ◯ Accumulator • common CPU components and their function: ◯ ALU (Arithmetic Logic Unit) ◯ CU (Control Unit) ◯ Cache • The function of the CPU as fetch and execute instructions stored in memory • How common characteristics of CPUs affect their performance: ◯ clock speed ◯ cache size ◯ number of cores • embedded systems:	Reading: • Regular use of on screen sources of information • Research and online reading and extracts Literacy: • Extended written responses across units • In depth research and referencing of sources • Use of spelling and grammar tools • Regular review of in class work focussed on level of written response • Modelling of appropriate level of written response Numeracy:	Regular exam questions and assessment against mark scheme criteria Regular opportunities to revisit previous tasks and improve based on feedback Verbal feedback on an individual basis Whole class feedback Extended end of unit assessment feedback	Three end of topic assessments One mock exam – Unit 1	1.1 – Systems Architecture 1.2 – Memory 1.3 – Storage 1.4 – Networks 1.5 – Topologies and protocols 1.6 – System Security 1.7 Systems Software 1.8 – Law, Ethics, Morals 2.1 – Algorithms 2.4 – Computational Logic 2.5 – Translators and facilities of languages 2.6 – Data Representation

◯ purpose of embedded systems ◯ examples of embedded systems. • the difference between RAM and ROM • the purpose of ROM in a computer system • the purpose of RAM in a computer system • the need for virtual memory • flash memory. • the need for secondary storage • data capacity and calculation of data capacity requirements • common types of storage: ◯ optical ◯ magnetic ◯ solid state • suitable storage devices and storage media for a given application, and the advantages and disadvantages of these, using characteristics: ◯ capacity ◯ speed ◯ portability ◯ durability ◯ reliability ◯ cost. • types of networks: ◯ LAN (Local Area Network) ◯ WAN (Wide Area Network) • factors that affect the performance of networks • the different roles of computers in a client-server and a peer-to-peer network	• Algebra – variables and data types • Logic and decision making • AND, OR, NOT • Conditional statements • Logic diagrams • Truth tables • Number conversions • Number systems • Encryption			

• the hardware needed to connect stand-alone computers into a Local Area Network: ◯ wireless access points ◯ routers/switches ◯ NIC (Network Interface Controller/Card) ◯ transmission media • the internet as a worldwide collection of computer networks: ◯ DNS (Domain Name Server) ◯ hosting ◯ the cloud • the concept of virtual networks. • star and mesh network topologies • Wifi: ◯ frequency and channels ◯ encryption • ethernet • the uses of IP addressing, MAC addressing, and protocols including: ◯ TCP/IP (Transmission Control Protocol/Internet Protocol) ◯ HTTP (Hyper Text Transfer Protocol) ◯ HTTPS (Hyper Text Transfer Protocol Secure) ◯ FTP (File Transfer Protocol) ◯ POP (Post Office Protocol) ◯ IMAP (Internet Message Access Protocol) ◯ SMTP (Simple Mail Transfer Protocol) • the concept of layers • packet switching. • forms of attack • threats posed to networks:				

◯ malware ◯ phishing ◯ people as the 'weak point' in secure systems (social engineering) ◯ brute force attacks ◯ denial of service attacks ◯ data interception and theft ◯ the concept of SQL injection ◯ poor network policy • identifying and preventing vulnerabilities: ◯ penetration testing ◯ network forensics ◯ network policies ◯ anti-malware software ◯ firewalls ◯ user access levels ◯ passwords ◯ encryption. • the purpose and functionality of systems software • operating systems: ◯ user interface ◯ memory management/multitasking ◯ peripheral management and drivers ◯ user management ◯ file management • utility system software: ◯ encryption software ◯ defragmentation ◯ data compression ◯ the role and methods of backup: ◯full ◯incremental.				

• how to investigate and discuss Computer Science technologies while considering: ◯ ethical issues ◯ legal issues ◯ cultural issues ◯ environmental issues. ◯ privacy issues. • how key stakeholders are affected by technologies • environmental impact of Computer Science • cultural implications of Computer Science • open source vs proprietary software • legislation relevant to Computer Science: ◯ The Data Protection Act 1998 ◯ Computer Misuse Act 1990 ◯ Copyright Designs and Patents Act 1988 ◯ Creative Commons Licensing ◯ Freedom of Information Act 2000. • computational thinking: ◯ abstraction ◯ decomposition ◯ algorithmic thinking • standard searching algorithms: ◯ binary search ◯ linear search • standard sorting algorithms: ◯ bubble sort ◯ merge sort ◯ insertion sort • how to produce algorithms using: ◯ pseudocode				

◯ using flow diagrams • interpret, correct or complete algorithms. • why data is represented in computer systems in binary form • simple logic diagrams using the operations AND, OR and NOT • truth tables • combining Boolean operators using AND, OR and NOT to two levels • applying logical operators in appropriate truth tables to solve problems • applying computing-related mathematics: ◯ + ◯ – ◯ / ◯ * ◯ Exponentiation (^) ◯ MOD ◯ DIV © OCR 2018 GCSE (9–1) in Computer Science 11 2.4 Computational logic Learners should have studied the following: • why data is represented in computer systems in binary form • simple logic diagrams using the operations AND, OR and NOT • truth tables • combining Boolean operators using AND, OR and NOT to two levels				

• applying logical operators in appropriate truth tables to solve problems • applying computing-related mathematics: ◯ + ◯ – ◯ / ◯ * ◯ Exponentiation (^) ◯ MOD ◯ DIV • characteristics and purpose of different levels of programming language, including low level languages • the purpose of translators • the characteristics of an assembler, a compiler and an interpreter • common tools and facilities available in an integrated development environment (IDE): ◯ editors ◯ error diagnostics ◯ run-time environment ◯ translators. Units • bit, nibble, byte, kilobyte, megabyte, gigabyte, terabyte, petabyte • how data needs to be converted into a binary format to be processed by a computer. Numbers • how to convert positive denary whole numbers (0–255) into 8 bit binary numbers and vice versa				

• how to add two 8 bit binary integers and explain overflow errors which may occur • binary shifts • how to convert positive denary whole numbers (0–255) into 2 digit hexadecimal numbers and vice versa • how to convert from binary to hexadecimal equivalents and vice versa • check digits. Characters • the use of binary codes to represent characters • the term 'character-set' • the relationship between the number of bits per character in a character set and the number of characters which can be represented (for example ASCII, extended ASCII and Unicode). Images • how an image is represented as a series of pixels represented in binary • metadata included in the file • the effect of colour depth and resolution on the size of an image file. Sound • how sound can be sampled and stored in digital form • how sampling intervals and other factors affect the size of a sound file and the quality of its playback: ◯ sample size ◯ bit rate ◯ sampling frequency. Compression				

• need for compression • types of compression: ◯ lossy ◯ lossless.				