January 2022

Computer Science Department – Curriculum Intent

KS3 Curriculum Intent			
Head of Department: Mr S Verma			
	Year 7	**Year 8**	**Year 9**
Taught over 3 terms	1. Introduction to coding through 'Kodu' 2. Microbit Programming and Game Development 3. Introduction to Python 4. Creating Apps via 'App Shed' 5. Games programming in Scratch	1. Graphic Design using Adobe Photoshop 2. HTML & Website Development/Games Programming in Scratch 3. Continuation with Python and 'While Loops/Searching' 4. Scratch and 'Edbot' or Cyberstart 5. Binary Data Representation	1. Introduction to Spreadsheets (Graphs, Formulae & Macros) 2. Code Combat 3. Python Next Steps (Lists and Procedures) 4. Code 4 life 5. Python and 'Edbot' 6. Cyber security (NCCE) 7. Cyberstart
Autumn Term	Introduction to coding through Kodu & Game development Introduction to Python Hour of Code in Dec A brief introduction to how the Interactive/Script mode works and a basic program that we work through eg. name calculator	Graphic Design using Adobe Photoshop - Staying safe online campaign. Continuation with Python and 'While Loops/Searching' Use of Pygame Use of Turtle and Tkinter/Pygame	Python – next steps Code 4 Life - students to complete the introduction levels and use elements of Python
Spring Term	Continuation of Python Project Microbits – Introduction to Physical programming using the BBC Microbits	Edbot programming using Scratch/Python HTML & Website Development	Gamemaker – Introduction to game making and introduction to Yoyo games Gamemaker. Students to complete the 'Tank' game and then create their own game using the skills that have been developed. Cyber security – Theory module through the NCCE module
Summer	App Shed- Create an app for Andriod and iOS. Games programming in Scratch	Computer crime & Cyber security Python – Use Python challenges and possibly look at RPG and Minecraft challenge. Binary Data	Speadsheets – Intro to using Excel, analysing data and creating Graphs Edbot/Robomaster – Introduction to robotics and programming with Python/Scratch

January 2022

Computer Science Department – Curriculum Intent

Overview of KS4 Curriculum		
Subject: GCSE Computer Science **Exam Board:** OCR		
	Year 10	**Year 11**
Autumn Term	**Unit 1 – Computer systems** 1.1.1 Architecture of the CPU 1.1.2 CPU performance 1.1.3 Embedded systems Introduction to Python and using Tkinter/Pygame - Students to use 'Learning Python' and work through the resources. **Unit 6 – Computational thinking, Algorithms & Programming** 2.1.1 Computational thinking 2.1.2 Designing, creating and refining algorithms	**Unit 4 - Network security & systems software** 1.4.1 Threats to computer systems and networks 1.4.2 Identifying and preventing vulnerabilities 1.5.1 Operating systems 1.5.2 Utility Software Python programming and regular challenges
Spring Term	**Unit 6 – Computational thinking, Algorithms & Programming (continued)** 2.1.3 Searching and sorting algorithms **Unit 2 – Data Representation** 1.2.1 Primary Storage 1.2.2 Secondary Storage 1.2.3 Units 1.2.4 Data Storage 1.2.5 Compression Python programming and regular challenges	**Unit 8 – Logic & Languages** 2.3.1 Defensive Design 2.3.2 Testing 2.4.1 Boolean Logic 2.5.1 Languages 2.5.2 IDE **Unit 5 – Ethical, legal, cultural and environmental impacts of digital tech**
Summer Term	**Unit 3 – Computer Network & Security Systems** 1.3.1 Networks & Topologies 1.3.2 Wired & Wireless networks, protocols and layers **Unit 7 – Programming – in preparation for Year 11 content** 2.2.1 Programming Fundamentals 2.2.2 Data Types 2.2.3 Additional programming techniques	Revision and Exam Practise – revisit Unit 7 & 2 Past paper packs to attempt

January 2022

Computer Science Department – Curriculum Intent

Overview of KS5 Curriculum				
Subject: A Level Computer Science **Exam Board:** OCR				
	Year 12		**Year 13**	
	Teacher A	**Teacher B**	**Teacher A**	**Teacher B**
Autumn Term	**Unit 1 – Components of a computer** 1.1.1 Structure and function of processors (a) The Arithmetic and Logic Unit; ALU, Control Unit and Registers (Program Counter; PC, Accumulator; ACC, Memory Address Register; MAR, Memory Data Register; MDR, Current Instruction Register; CIR). Buses: data, address and control: how this relates to assembly language programs. (b) The Fetch-Decode-Execute Cycle; including its effects on registers. (c) The factors affecting the performance of the CPU: clock speed, number of cores, cache. (d) The use of pipelining in a processor to improve efficiency. (e) Von Neumann, Harvard and contemporary processor architecture. 1.1.2 Types of processor (a) The differences between and uses of CISC and RISC processors. (b) GPUs and their uses (including those not related to graphics). (c) Multicore and Parallel systems. 1.1.3 Input, output and storage (a) How different input, output and storage devices can be applied to the solution of different problems. (b) The uses of magnetic, flash and optical storage devices. (c) RAM and ROM. (d) Virtual storage.	**Unit 6 – Data Types** 1.4.1 Data Types a) Primitive data types, integer, real/ floating point, character, string and Boolean. b) Represent positive integers in binary. c) Use of sign and magnitude and two's complement to represent negative numbers in binary. d) Addition and subtraction of binary integers. e) Represent positive integers in hexadecimal. f) Convert positive integers between Binary Hexadecimal and denary. g) Representation and normalisation of floating point numbers in binary. h) Floating point arithmetic, positive and negative numbers, addition and subtraction. i) Bitwise manipulation and masks: shifts, combining with AND, OR, and XOR. j) How character sets (ASCII and UNICODE) are used to represent text **Programming- Intro to C# and Unity** Students are introduced to C# and Unity and follow the tutorials issues by the teacher as well as using the online learning hub.	**Unit 4 – Exchanging data** 1.3.2 Databases a) Relational database, flat file, primary key, foreign key, secondary key, entity relationship modeling, normalisation and indexing. See appendix 5g. b) Methods of capturing, selecting, managing and exchanging data. c) Normalisation to 3NF. d) SQL - Interpret and modify. See appendix 5d. e) Referential Integrity. f) Transaction processing, ACID (Atomicity, Consistency, Isolation, Durability), record locking and redundancy. **Unit 10 – Computational thinking** 2.2.2 Computational methods a) Features that make a problem solvable by computational methods. b) Problem Recognition. c) Problem Decomposition. d) Use of divide and conquer. e) Use of abstraction. f) Learners should apply their knowledge of: • backtracking • data mining • heuristics • performance modelling • pipelining • visualisation to solve problems	**Unit 8 – Boolean Algebra** 1.4.3 Boolean Algebra a) Define problems using Boolean logic. See appendix 5d. b) Manipulate Boolean expressions, including the use of Karnaugh maps to simplify Boolean expressions. c) Use the following rules to derive or simplify statements in Boolean algebra: De Morgan's Laws, distribution, association, commutation, double negation. d) Using logic gate diagrams and truth tables. See appendix 5d. e) The logic associated with D type flip flops, half and full adders. **Unit 9 Legal, Moral & ethical issues** 1.5.2 Ethical, moral and cultural issues a) The individual (moral), social (ethical) and cultural opportunities and risks of digital technology: • Computers in the workforce • Automated decision making • Artificial intelligence • Environmental effects • Censorship and the Internet • Monitor behavior • Analyse personal information • Piracy and offensive communications • Layout, colour paradigms and character sets.

January 2022

Computer Science Department – Curriculum Intent

	Unit 2 – Software 1.2.1 Operating Systems a) The need for, function and purpose of operating systems. b) Memory management (paging, segmentation and virtual memory). c) Interrupts, the role of interrupts and Interrupt Service Routines (ISR), role within the fetch decode execute cycle. d) Scheduling: round robin, first come first served, multi-level feedback queues, shortest job first and shortest remaining time. e) Distributed, embedded, multi-tasking, multi-user and real time operating systems. f) BIOS. g) Device drivers. h) Virtual machines, any instance where software is used to take on the function of a machine, including executing intermediate code or running an operating system within Another. 1.2.2 Applications Generation a) The nature of applications, justifying suitable applications for a specific purpose. b) Utilities. c) Open source vs closed source. d) Translators: interpreters, compilers and assemblers. e) Stages of compilation (Lexical analysis, Syntax analysis, Code generation and Optimisation). f) Linkers and loaders and use of libraries 1.2.3 Software Development			**NEA Project** Analysis & Design sections to be completed. Development of the solution to have started and aiming to complete by January.

January 2022

Computer Science Department – Curriculum Intent

	a) The nature of applications, justifying suitable applications for a specific purpose. b) Utilities. c) Open source vs closed source. d) Translators: interpreters, compilers and assemblers. e) Stages of compilation (Lexical analysis, Syntax analysis, Code generation and Optimisation). f) Linkers and loaders and use of Libraries			
Spring Term	**<u>Unit 3 – Software Development</u>** 1.2.4 Types of Programming Language a) Need for and characteristics of a variety of programming paradigms. b) Procedural languages. c) Assembly language (including following and writing simple programs with the Little Man Computer instruction set). See appendix 5d. d) Modes of addressing memory (immediate, direct, indirect and indexed). e) Object-oriented languages with an understanding of classes, objects, methods, attributes, inheritance, encapsulation and polymorphism.	**<u>Unit 7 – Data Structures</u>** 1.4.2 Data Structures a) Arrays (of up to 3 dimensions), records, lists, tuples. b) The following structures to store data: linked-list, graph (directed and undirected), stack, queue, tree, binary search tree, hash table. c) How to create, traverse, add data to and remove data from the data structures mentioned above. **<u>Unit 12 – Algorithms</u>** 2.3.1 Algorithms a) Analysis and design of algorithms for a given situation. b) The suitability of different algorithms for a given task and data set, in terms of execution time and space. c) Measures and methods to determine the efficiency of different algorithms, Big O notation. (Constant, linear, polynomial, exponential and logarithmic complexity) d) Comparison of the complexity of algorithms. e) Algorithms for the main data structures, (Stacks, queues, trees, linked	**<u>Unit 11 – Programming techniques</u>** 2.2.1 Programming techniques a) Programming constructs: sequence, iteration, branching. b) Recursion, how it can be used and compares to an iterative approach. c) Global and local variables. d) Modularity, functions and procedures, parameter passing by value and by reference. e) Use of an IDE to develop/ debug a program. f) Use of object oriented techniques **Revision – recap of earlier modules for refreshing knowledge.**	**<u>NEA Project</u>** Development and Testing to aid development ocumentation to be marked and completed during this term & moderation finalised.

Computer Science Department – Curriculum Intent

		lists, depth-first (post-order) and breadth-first traversal of trees). f) Standard algorithms (Bubble sort, insertion sort, merge sort, quick sort, Dijkstra's shortest path algorithm, A* algorithm, binary search and linear search).		
Summer Term	**Unit 5 – Networks & Web Technologies** 1.3.3 Networks a) Characteristics of networks and the importance of protocols and standards. b) The internet structure: • The TCP/IP Stack. • DNS • Protocol layering. • LANs and WANs. • Packet and circuit switching. c) Network security and threats, use of firewalls, proxies and encryption. d) Network hardware. e) Client-server and peer to peer. 1.3.4 Web Technologies- a) HTML, CSS and JavaScript. See appendix 5d. b) Search engine indexing. c) PageRank algorithm. d) Server and client side processing.	**NEA Project** Intro to the project and start on Analysis section. Go through how to document and create the project. Students should have brainstormed suitable ideas and gone through the 'Project Guidance' to ensure that their project is suitable.	**Revision and recap of earlier modules from Year 12** **Past papers to be completed**	**Revision and recap of earlier modules from Year 12**

January 2022

Computer Science Department – Curriculum Intent

Curriculum Rationale:

It is the belief of Urmston Grammar Computer Science department that Computer Science develops children's thinking for logic and problem solving which is one of the major skills needed in the modern job market and at university. With that in mind we have created a Computer Science curriculum that encompasses problem solving and real life programming with languages that are used in industry eg Python and C#. We support the theory with physical programming of edbots, microbits and dji drones that we have in the department. The aims of this curriculum are to enable learners to develop:

- An understanding and ability to apply the fundamental principles and concepts of computer science, including: abstraction, decomposition, logic, algorithms and data representation
- The ability to analyse problems in computational terms through practical experience of solving such problems, including writing programs to do so
- The capacity to think creatively, innovatively, analytically, logically and critically
- The capacity to see relationships between different aspects of computer science
- Mathematical skills.

At KS3 students work hard to rapidly develop programming skills and knowledge. With a number of students beginning a programming language for the first time, we are delighted that our students make excellent progress and many choose to continue onto GCSE Computer Science as well as using the skills learnt in other subjects eg Using software packages such as Office, Adobe and Python.

At KS4-5 students follow the OCR syllabus for Computer Science and this entails theory modules as well as encouraging students to develop their understanding and application of the core concepts in computer science. Students analyse problems in computational terms and devise creative solutions by designing, writing, testing and evaluating programs.

Careers:

The ability to program in Python, C# and HTML which is taught at KS3-5 can lead to the following careers. Analytical skills such as debugging programs and recognising how to fix them is a skill that is valued amongst employers, as well as the below career fields in Computer Science.

Application analyst
Applications developer
Cyber security analyst
Data analyst
Database administrator
Forensic computer analyst
Game designer
Games developer
IT consultant
Software engineer
Systems analyst
UX designer
Web designer
Web developer