DRAYTON MANOR HIGH SCHOOL

Faculty Curriculum Map – 2025/26

Faculty: Computer Science/IT

Year	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
7	**Classoos – Topic 1** **Using computers Safely effectively and Responsibly** - File Management - Social networking - Keeping your data safe - Using Email - Searching the Web - Final assessment	**Programming essentials in Scratch – part I** - Introduction to programming and sequencing - Sequence and variables - Selection - Operators - Count-controlled iteration - Problem-solving	Modelling data using **spreadsheet**s	Artificial Intelligence and Machine Learning	Programming essentials in Scratch – part II	Using media – Gaining support for a cause
8	**Classoos – Topic 3** **Computational thinking** - Logical Thinking & Logic Gates - Algorithmic Thinking - Abstraction - Decomposition - Searching Algorithms - Sorting Algorithms	**Introduction to Python** - Inputs, outputs and sequences - Variables - Data Types - Arithmetic - Branching - looping	**Networks** from semaphores to the internet	**IT** - Word Processing – letter - Web search - Presentation	Developing for the Web	Artificial Intelligence and Machine Learning
9	**EduBlocks Python** - Comparing Scratch & Python - Turtle + Iteration - User Input & Data Types - Variables - Functions	Artificial Intelligence and Machine Learning	PhotoPea - edit and manipulate your images	Understanding Computers		Applying programming skills with physical computing

	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
10 Computer Science	**1.2 Memory and storage** • Primary storage (Memory) • Secondary storage • Units • Data storage • Compression **2.2 - Programming - BBA** • Programming fundamentals – the three constructs • Data types **End-of-topic test (1.2 and Programming)**	**1.1 - Systems architecture** • Architecture of the CPU • CPU performance • Embedded systems **2.2 - Programming** • Additional programming techniques **End-of-topic test (1.1 and Programming)**	**1.3 - Computer networks** **End-of-topic test (1.3 and Programming)** Dedicated programming lessons	**1.4 - Network security** End-of-topic test (1.4 and Programming) Dedicated programming lessons **1.5 - System software** • 1.5.1 Operating systems • 1.5.2 Utility software **End-of-topic test (1.5)**	**1.6 - Impacts of digital technology** • Ethical, legal, cultural and environmental impact • Legislation relevant to Computer Science **End-of-topic test (1.6)**	**2.1 Algorithms** • Computational thinking • Designing, creating and refining algorithms • Searching and sorting algorithms End-of-topic test (2.1) **End of Year exam**

	Autumn 1	**Autumn 2 (4 weeks)**	**Spring 1**	**Spring 2**	**Summer 1**	**Summer 2**
11 Computer Science	**Baseline Test** **1.5 - System software** • 1.5.1 Operating systems • 1.5.2 Utility software **End-of-topic test (1.5)** **1.6 - Impacts of digital technology** • Ethical, legal, cultural and environmental impact • Legislation relevant to Computer Science **End-of-topic test (1.6)** **2.2 - Programming** Additional programming techniques - File handling - String manipulation - SQL	**2.4 - Boolean logic** • Logic diagrams using the operators AND, OR and NOT • Truth tables **End-of-topic test (2.4)** **2.3 - Producing robust programs** • Defensive design • Testing **End-of-topic test (2.3)** **2.5 - Programming languages and IDE** • Editors • Error diagnostics • Run-time environment • Translators **End-of-topic test (2.5)** **December Mock**	**Tailored Revision**	**Tailored Revision**	**Tailored Revision**	

	Autumn 1		Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
11IT	**R050 –** Exam IT in the Digital World Exam unit	**RO60 –** Data manipulation using spreadsheets Coursework	**R070 -** Using Augmented Reality to present information Coursework **Task 1** – Designing the Augmented Reality (AR) model prototype **Task 2** – Creating the Augmented Reality (AR) model prototype **Task 3** – Test and review the Augmented Reality (AR) model prototype **Submit RO60 coursework** **R050 –** Exam IT in the Digital World **December Mock Revision**	**R070 -** Using Augmented Reality to present information Coursework **Task 2** – Creating the Augmented Reality (AR) model prototype **Task 3** – Test and review the Augmented Reality (AR) model prototype **Tailored Revision**	**R070 -** Using Augmented Reality to present information Coursework **Task 3** – Test and review the Augmented Reality (AR) model prototype **Submit RO70 coursework** **Tailored Revision**	**R050** – IT in the Digital World Final Exam **Tailored Revision**	
	TA5 – Digital Communication • 5.1 Types • 5.2 Software • 5.3 Digital devices • 5.4 Distribution channels • 5.5 Audience demographics **TA5 Topic Test** **TA6 –** Internet of Everything • 6.1 Use of IoE • 6.2 Application areas in everyday life **TA6 Topic Test**	**Task 1 –** Planning and designing a spreadsheet solution **Task 2 –** Creating a spreadsheet solution **Task 3 –** Testing and evaluating a spreadsheet solution					

	Autumn 1	Autumn 2		Spring 1	Spring 2	Summer 1	Summer 2
12 Computer Science	**BBA x3**	**SEV x2**					
	Component 02 - Algorithms and programming 2.2.1 Programming techniques a) Programming constructs: sequence, iteration, branching. f) Use of object oriented techniques. **End-of-topic test (2.2.1)** • Mini Projects ➢ Defold ➢ JavaScript ➢ Database **1.4.2 Data Structures** (a) Arrays, records, lists, tuples. (b) 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 **End-of-topic test (1.4.2)**	**1.3.4 Web Technologies** (a) HTML, CSS and JavaScript. (b) Search engine indexing. (c) PageRank algorithm. (d) Server and client-side processing. **End-of-topic test (1.3.4)** **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. **End-of-topic test (1.4.1)**		1.3.2 Databases 1.3.1 Compression, Encryption and Hashing 1.3.3 Networks	1.2 Software and software development	NEA Project lessons – Analysis – 10 marks NEA Project lessons – Design 15 marks 1.5 Legal, moral, cultural and ethical issues UCAS exam	

	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
13 Computer Science	**1.4.2 Data Structures** (a) Arrays, records, lists, tuples. (b) 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 **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. **NEA Design**	**2.3 Algorithms** (a) design of algorithms (b) suitability of different algorithms (c) efficiency of different algorithms, Big O notation (d) Comparison of the complexity of algorithms. (e) Algorithms for the main data structures, (f) Standard algorithms (bubble sort, insertion sort, merge sort, quick sort, Dijkstra's shortest path algorithm, A* algorithm, binary and linear search). **1.2.4 Types of Programming Language** (c) Assembly language (including following and writing simple programs with the Little Man Computer instruction set). b) Recursion, how it can be used and compares to an iterative approach **NEA Development**	**Component 01 - Computer systems** **1.2 Software and software development** 1.2.2 Applications Generation 1.2.3 Software Development **2.1 Elements of computational thinking** • 2.1.1 Thinking abstractly • 2.1.2 Thinking ahead • 2.1.3 Thinking procedurally • 2.1.4 Thinking logically 2.1.5 Thinking concurrently NEA Submission	**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: • backtracking • data mining • heuristics • performance modelling • pipelining • visualisation to solve problems	**Tailored Revision**	

	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
IT Carousel	**AI and Machine Learning** ➢ What is AI? ➢ Machine learning ➢ Ethics of AI ➢ Image recognition ➢ Turing tests and chatbots ➢ Rate my review					