MAIDSTONE GRAMMAR SCHOOL
FOUNDED 1549

Computer Science Curriculum Outline

This outline provides a long-term overview of the knowledge and skills developed in this subject. More detailed short- and medium-term schemes of work, not published here, are available by contacting the Computing Department.

	Term 1	Term 2	Term 3	Term 4	Term 5	Term 6
Year 13	**AQA COMPUTER SCIENCE A-level 7517** **Knowledge & Skills:** ** some of these topics will be covered in Y12 Term 6 • Vectors • Logic programming • Functional programming • Regular expressions • BNF • Optimising algorithms (Dijkstra's shortest path algorithm) • Big Data • Turing Machines • The role of the Operating System	**AQA COMPUTER SCIENCE A-level 7517** **Knowledge & Skills:** NEA Project development time.	**AQA COMPUTER SCIENCE A-level 7517** **Knowledge & Skills:** • AQA Preliminary material & skeleton program investigation. • Applications and effects of using computers. • Computing laws.	**AQA COMPUTER SCIENCE A-level 7517** **Knowledge & Skills:** • AQA Preliminary material & skeleton program mini-mock assessment. • NEA VIVAs • Specification Review & RAG rating • Targeted Revision & Practice of Examination Questions (focus identified from specification review and mock examinations)	**AQA COMPUTER SCIENCE A-level 7517** **Knowledge & Skills:** • Timed Past Paper Practice Questions • Timed Programming Challenges (Section B) • AQA Preliminary material & skeleton code continued practice	
Year 12	**AQA COMPUTER SCIENCE A-level 7517** **Knowledge & Skills:** • Introduction to Computer Science. • Computational Thinking • Boolean Algebra. • Binary integer arithmetic (addition, subtraction and multiplication).	**AQA COMPUTER SCIENCE A-level 7517** **Knowledge & Skills:** • Windows programming Representing text. • Architecture & assembly language programming. • Representing graphics and sound. • Encryption.	**AQA COMPUTER SCIENCE A-level 7517** **Knowledge & Skills:** • Object Oriented programming. • Comparing algorithms and Big O notation. • Relational databases. • Normalising to 3NF.	**AQA COMPUTER SCIENCE A-level 7517** **Knowledge & Skills:** • Networks 1 • UCAS exam Preliminary • Materials	**AQA COMPUTER SCIENCE A-level 7517** **Knowledge & Skills:** • FSM • Representing real numbers in Binary.	**AQA COMPUTER SCIENCE A-level 7517** **Knowledge & Skills:** • System Life cycle • Introduction to the NEA • NEA Proposals & Analysis *** topics not already covered but would be helpful for the*

	• Data structures 1 (stacks, queues, dictionaries, hashing).					*NEA. These are taken from the list in Yr13 term 1*
	Term 1	Term 2	Term 3	Term 4	Term 5	
Year 11	**GCSE Computer Science J277** **Knowledge:** • Hardware • Memory (inc. Virtual memory) • Storage **Knowledge & Skills**: • Defensive programming • **Programming Project 5**	**GCSE Computer Science J277** **Knowledge**: • Programming languages and translation. • System Architecture CPU performance. • Embedded Systems. **Skills**: • Using IDE to demonstrate translation. • Writing assembly language programs.	**GCSE Computer Science J277** **Knowledge & Skills**: • Software • Operating Systems • Utility programs	**GCSE Computer Science J277** **Knowledge & Skills**: • Specification Review & RAG rating • Targeted Revision & Practice of Examination Questions (focus identified from specification review and mock examinations)	**GCSE Computer Science J277** **Knowledge & Skills**: • Timed Past Paper Practice Questions • Timed Programming Challenges • Topic-Based Revision as Required	
	Term 1	Term 2	Term 3	Term 4	Term 5	Term 6
Year 10	**GCSE Computer Science J277** **Knowledge**: • Formal algorithm design (revision from previous lessons). • Defensive design. • Testing and text data. • **Program project 3.** **Skills**: • Write algorithms in flowcharts and Pseudo code (concentrating on the OCR ERL). • Be able to write robust programs. • Test a program against the objectives using a range of test data.	**GCSE Computer Science J277** **Knowledge**: • Networks and topologies. • Wired and wireless networks, protocols and layers. • Standard Algorithms (Linear and Binary searching & Bubble sort). • Threats to computer systems and networks. • Identifying and preventing vulnerabilities. **Skills**: • Be able to answers exam questions on networks, the Internet and risks • Understand and write algorithms for linear search and binary search. • Write bubble sort programs.	**GCSE Computer Science J277** **Knowledge**: • Standard sorting algorithms cont. (Insertion and Merge sorts). • Threats to computer systems and networks (if not covered in Term 2). • Identifying and preventing vulnerabilities (if not covered in Term 2). **Skills**: • Trace algorithms for insertion sort. • Draw diagrams to explain merge sort. • Consider the best method for searching and sorting for a given situation.	**GCSE Computer Science J277** **Knowledge**: • Build upon the KS3 data handling. • Understand and Use DML and DDL. **Skills**: • Create an external database using DDL commands. • Use CRUD to manipulate the data within the database (single table). • Write simple and complex SELECT Queries • Be able to answer exam questions using SELECT query.	**GCSE Computer Science J277** **Knowledge**: **Programming Project 4.** **Skills**: • Creating a structured program for a given task, using the system life cycle.	**GCSE Computer Science J277** **Knowledge**: • Computer Legislation • Migration to a second HLL and language specific structurers **Skills**: • Understand the copyright, design and patent Act; Freedom of Information Act; Computer Misuse Act and the data protect Act (GDPR). • Experience using another HLL, using sequence, selection and iterations. • Understand why there are different HLL.

	Term 1	Term 2	Term 3	Term 4	Term 5	Term 6
Year 9	**GCSE Computer Science J277** **Knowledge**: • Programming fundamentals • Strong data types • Units of data storage • Representing numbers **Skills**: • Use a HLL IDE to write programs using sequence and selections (following from Yr8). • Understand variable and constants and their strong data types. • Be able to use units appropriately. • Convert between binary, denary and hexadecimal. • Perform binary addition and identify overflow. • Understand simple data checking using parity. • Use shifts to multiply and divide by 2.	**GCSE Computer Science J277** **Knowledge**: • Computational thinking • Representing Characters • Test plans and appropriate test data (actual, boundary and erroneous) • Understand Logic Gates (ANO, OR & NOT) • Draw Logical systems and circuits. **Skills**: • Writing Programming– selection using CASE and iteration (conditional and unconditional loops). • Use computational thinking to solve problems. • Understand how characters are represented within a computer system. Drawing a logic circuit to represent a system and representing the logic circuit as a Truth Table **Guided Programming Project 1**	**GCSE Computer Science J277** **Knowledge**: • Understand the representation of data within a computer system – graphics, • Bit mapped graphics Compression: Lossy and Lossless • Compression techniques: RLE & dictionary • Design programs using Pseudo code and ERL. • Find errors using dry runs and trace tables. **Skills**: • Understand how graphics are stored within a Computer System. • Explain compression and the different types. • Use arrays (1D and 2D).	**GCSE Computer Science J277** **Knowledge**: • Computational Thinking • Defining and refining algorithms • The IDE **Skills**: • Understand and use abstraction, decomposition and algorithmic thinking. • Represent algorithms in Pseudo Code (OCR ERL) • Dry run algorithms and create trace tables. • Use the UDE effectively to find logic and syntax errors. • Use string manipulation commands.	**GCSE Computer Science J277** **Knowledge**: • Representing sound • **Programming Project 2** **Skills**: • Understand how sound is represented with a computer system. • Understand what affects the sound quality. • Following the systems life cycle to create a computer system.	**GCSE Computer Science J277** **Knowledge**: • Structured programming using subroutines and Functions. • Variable Scope – local and global • Parameters and passing parameters to subroutines and functions • Reading and writing to text files. **Skills**: • Understand and write programs with subroutines and functions. • Understand the difference between a subroutine and a function. • Understand the advantages of structured programming.
	Term 1	Term 2	Term 3	Term 4	Term 5	Term 6
Year 8	**Programming Part 2** **Knowledge**: This topic builds upon the programming done in Year 7. • Introduction to computational thinking	**Microbits** **Knowledge**: • Using the Microbit system and drag and drop block programming to program and implement a	**Information and Data** **Knowledge**: Know how data is stored within a computer system • Denary & Binary number system • ASCII	**Making Effective Presentations** **Knowledge**: Present finding using Word as a formal report • Style sheets • Table of contents	**Writing Reports Effectively** **Knowledge**: Be able to research and digital present their findings to a variety of audiences and purposes.	**Creating web-based presentations** **Knowledge**: For a given topics, students will be able to: • Research

	— Abstraction — Decomposition — Algorithms • Creating flowcharts to solve a problem. • Introduction to textual programming using Small Basic IDE **Skills**: • Algorithm Design using Flowcharts (following from Yr7 Flowol projects). • Variables and Data Types (Integers and strings). • Programs and flowcharts demonstrating Sequence. • Programs and flowcharts Selection (IF THEN ELSE) Inc. nested IF statements. • Programs and flowcharts Iteration (Conditional and non-conditional). • Creating a Small Basic program from a flowchart design. • Finding and correcting programming err or.	variety of tasks highlighting IPO systems. • Embedded systems. **Skills**: • To develop algorithms and programs using iteration and selection. • To experiment with graphical and text-based programming languages. • To be able to explain input and output devices, hardware and software. • To apply understanding to writing algorithms and programming using Microbit. • To extend the Microbits by adding additional sensors and actuators.	• Encryption (symmetric substitution) • Programming Caesar Shift in Small Basic **eSafety**: understand the risks associated with: • Social Networks • Identity Protection **Skills**: • Convert denary to binary. • Convert binary to denary. • Understand use of binary to represent characters in ASCII • Understand the use of certain encryption types: — Caesar Cypher — Cuniform — Keyword Cypher — Morse code — Semaphore — Hash [optional] — Use of frequency tables to decode • Be able to code simple encryption using small basic.	• Cross referencing (citations and Bibliography) • Captioning pictures • Content appropriate to purpose and audience Present findings using PowerPoint (Using PowerPoint correctly) • Themes • Styles • Layout • Animations • Content appropriate to purpose and audience **Skills**: • Use the Slide master. • Set a custom background. • Apply a style. • Layout the presentation for viewing as part a talk. • Use animation as appropriate for a talk (bullet points and images appear as needed). • Use the notes section for talk content. • Be able to extract information from research and adapt for talk.	**Skills**: • Be able to use Style Sheets (body, heading 1, heading 2 etc). • Be able to create a table of contents. • Be able to insert and position images with text wrap. • Be able to Caption images. • Be able to apply header and footer as appropriate. • Be able to insert Citations. • Be able to insert and manage sources. • Be able to create a bibliography from sources. *** Using current research tools including AI (such as ChatGPT)*	• Use of grouping to produce segments • Inserting images and Animation • Use of Templates and Themes Computational Thinking • Analysis and decomposition • Sequencing **Skills**: • Use of templates and styles. • Targeted research procedures. • Be able to group text and images. • Be able to group images in different appropriate structures. • Be able to evaluate their own and others digital products.
	Term 1	Term 2	Term 3	Term 4	Term 5	Term 6
Year 7	**Introduction To Our Systems** **Knowledge**: • Passwords, Log in • File organisation (local and OneDrive) • Social effects of ICT — Copyright — Social Media — Potential pitfalls • Create a publication for • Purpose and audience	**Programming with a graphical Interface** **Knowledge**: • Game documentation & evaluation — Create and control sprites. — Code routines to interact with other sprites and environments.	**Modelling** **Knowledge**: • Design, use and evaluate computational models the state and behaviour of real-world problems and physical systems. • Use a variety of models to answer what if questions.	**Control Systems** **Knowledge**: • Investigate “real” control systems • Understand actuators and sensors • Draw and follow flowcharts Mimic project — Zebra crossing & Simple crossing.	**Data Handling** **Knowledge**: • Murder Mountain (Searching) — Using an existing database to perform simple and complex searches (AND, OR and NOT). — Use an existing database to answer questions.	**Writing programs with a textual language** **Knowledge**: Using a text based programming language to implement a variety of tasks • Write programs using sequence of instructions • Write programs that output information in an appropriate form

	Skills: • Logging into VLE and email. • Finding resources. • Submitting resources. • OneDrive (and the mobile apps). • Using OneNote to present information. • Using media to inform a target audience. • Using TEAMS.	— Understand variables and using them effectively. — Designing a maze based game. **Skills**: • Sequencing of instructions • Creating events • Create Clones • Passing parameters (Broadcasting) • Selection (IF) • Iteration (FOREVER, FOREVER IF) • Interactions	• Use formulae and variables to build models for real events. • Analyse and present the results of the model. **Skills**: • Understand the concept of a model. • Use existing models by changing data and evaluating the results. • Write appropriate rules, busing simple arithmetic and aggregation functions. • Plan and create their own model to represent a real situation.	— Traffic lights over a bridge & Pelican crossing. — Lighthouse/ crib mobile. — Ferris wheel / drinks machine. — Greenhouse / automated house / trains. **Skills**: • Understand IPO. • Understanding control systems and relate to "real world" situations. • Investigate different hardware. • Flowcharts: Sequence, input/output and Decisions. • Using algorithms (flowcharts) to control physical systems. • Identify the advantages and disadvantages of control systems.	• Top Trumps (Creating) — Design a flat-file record structure. — Populate a database, checking for data errors. • Data Handling laws **Skills**: • Boolean Logic (AND, OR and NOT) Venn Diagrams. • Design, Create and develop data capture forms; Complete and evaluate suitability. • Recognise different types of data: text; number; instruction. • Design, create and refine data structures; Simple validation and input masks. • Populate databases. • Queries; AND, OR and NOT; Wildcard; Use database to answer natural language questions.	• Write programs that make selections • Write programs that repeat block of code **Skills**: • Variables and Data Types (Integers and strings). • Programs and flowcharts demonstrating Sequence. • Programs and flowcharts Selection (IF THEN ELSE) Inc. nested IF statements. • Programs and flowcharts Iteration (Conditional and non-conditional).

Key/Legend/Notes: