CURRICULUM AND ASSESSMENT PLAN

Alevel CS YEAR 12

ORMISTON SWB ACADEMY

INTENT

Our faculty mission is to equip students with a robust understanding of computer science principles, practical programming skills, and the ability to apply computational thinking to solve real-world problems. We aim to develop students who are not only proficient users of technology but also creative thinkers and informed digital citizens. The curriculum is designed to challenge and **Develop Computational Thinking**: Teach students how to think logically, solve problems efficiently, and understand the principles of computer systems and networks, Advance P**rogramming Proficiency**: Introduce students to programming languages such as Python, ensuring they understand basic syntax, control structures, and algorithms. Mature **Digital Literacy skills**: Ensure students are competent and safe users of technology, understanding online safety, data privacy, and the ethical implications of technology.

Prior learning	GCSE *Computer Systems*, which explores hardware, networks, security, and ethical issues, and *Computational Thinking, Algorithms, and Programming*, which focuses on coding principles, logic, and problem-solving. Students develop analytical and programming skills, preparing for exams that assess both theoretical knowledge and practical application.
Conscious curriculum links	The **A-Level Computer Science curriculum** integrates key computational concepts with broader educational themes, fostering problem-solving, analytical thinking, and ethical considerations in technology. It covers programming, algorithms, data structures, cybersecurity, and the societal impacts of computing, ensuring students develop both technical expertise and an awareness of real-world applications
Extra-curricular	Bebras Computing Challenge is an international competition that promotes computational thinking among students aged 6 to 18. Organized in over 80 countries.

	AUTUMN 1	AUTUMN 2	SPRING 1	SPRING 2	SUMMER 1	SUMMER 2
TOPIC/KNOWLEDGE	**The characteristics of contemporary processors, input, output and storage devices** 1.1.3 Input/Output and storage 1.1.1 Structure and function of the processor 1.2.1 Operating systems **2.2 Problem solving and programming** **Selection** **Iteration** Programming techniques	**Types of software and the different methodologies used to develop software** 1.2.1 System software 1.2.2 Application generation 1,2.3 Software Development 1.2.4 Types of programming language **2.2 Problem solving and programming** Basic data structures Programming techniques	**How data is exchanged between different systems** 1.3.1 compression, encryption and hashing 1.3.2 Databases 1.3.3 Networking 1.3.4 web technologies **2.2 Problem solving and programming** Basic data structures Programming techniques	**How data is represented and stored within different structures, Different algorithms that cab b applied to these structures** 1.4.1 data types 1.4.2 Data structures 1.4.3 Boolean algebra **Understanding** **2.2 Problem solving and programming** Basic data structures Subroutines Object oriented programming	**Pygame Zero** **Pygame** **Programming techniques** **Understand what is meant by computational Thinking** 2.1.1 Thinking abstractly 2.1.2 Thinking procedurally 2.1.4 Thinking logically 2.1.5 Thinking concurrently	**Legal, Moral, Cultural and ethical issues** 1.5.1 Computer related legislation 1.5.2 Moral and ethical issues **The use of algorithms to describe problems and standard algorithms** 2.3.1 Algorithms **Unit 3 NEA** 3.1 Analysis of the problem 3.2 Design of the solution 3.3 Develop the solution
SKILLS	A comprehensive list of all Alevel Computer Science skills can be found at: https://www.ocr.org.uk/images/170844-specification-accredited-a-level-gce-computer-science-h446.pdf					
ASSESSMENT	CS- 1.1 Contemporary Processors, I/O and Storage Devices & 2.2 Problem solving and programming – Basics CS – Fetch-decode & execute effect on registers CS – Storage devices, and CPU performance	CS – 1.2 Software & Software Development & 2.2 Problem solving and programming - Selection, Iteration & Basic data Structures CS – Bios, Virtual machines & scheduling and Interrupts	CS – 1.3 Exchanging Data & 2.2 Problem solving and programming – Subroutines & IDE CS – Lossy Vs Lossless, symmetric & asymmetric encryption CS – Characteristics of networks and the protocols	CS – 1.4 Data Types, data structures and algorithms & 2.1 Elements of computational thinking CS – Primitive data types, integers, real/floating point	• Y12 Mock exams	CS- 1.5 Legal, moral, cultural and ethical issues CS – Know all computing legislation CS – Individual moral, social, ethical and cultural opportunities
VOCAB	**Key terminology:** Hardware Secondary storage Optical Magnetic Solid state Storage Characteristic Estimate Overheads	**Key terminology:** System Software, Operating System (OS), Virtual Machine, Utility Programs, Device Drivers, Multitasking, Time Slice, Processor Management, User Interface, Graphical User Interface, Command Line Interface, Peripheral Management, File Management, Utility Program, Operating System, Defragmentation, Compression, Lossy, Lossless, Encryption, Backup, Incremental, Full	**Key terminology:** • baud rate • bit rate • bandwidth • latency • protocol, • physical star topology • logical bus network topology	**Key terminology:** • Bit Nibble Byte Kilobyte Megabyte Gigabyte Terabyte Petabyte Binary Denary Conversion Transistor Bit Binary Denary Left • Mid • Right • Addition • Subtraction • Division • modular division • integer division • negation • AND • OR • NOT • Greater than • Less than • Equal	**Key terminology:** Requirements Interaction Prototype Agile Abstraction Solution structure documented design iterative debugging feedback critical path testing errors boundary erroneous evaluation	• **Key terminology:** Ethical issues, Legal issues, Cultural issues, Environmental issues, Privacy issues, Stakeholder, Open source software, Proprietary software, Data Protection Act, Computer Misuse Act, Copyright Designs and Patents Act, Creative commons licensing, Freedom of Information Act

READING SKILLS

Reading Skills

Pupils develop critical reading skills by interpreting and understanding complex programming documentation. Pupils learn to extract relevant information, understand technical jargon, and comprehend detailed instructions. Computing emphasizes the importance of clear and precise writing. Pupils engage in writing code comments, documentation, and project reports, which require them to explain their thought processes and solutions in a coherent manner. This practice enhances their ability to articulate complex ideas. Digital literacy skills pupils will understand how to navigate and utilise various software tools, platforms, and online resources which is crucial in today's digital world. This competence supports their ability to research, learn independently, and engage with digital content critically and safely.

CAREERS LINKS

CS Career Developments: Software Engineer, IT Support Specialist, Computer Systems Analyst, AI Engineer, Robotics Engineer. Developments: Software Engineer, Cyber Security Analyst, Cloud Solutions, Games developer, Robotics

CORE

In computing, these values support problem-solving, persistence in debugging, effective project management, and striving for high-quality work. The curriculum reinforces personal development, emotional wellbeing, and responsible online behaviour, helping pupils build resilience and strong character both digitally and personally.

SUPPORTING STUDENTS AT HOME

we support pupils' learning at home through a combination of structured homework tasks and knowledge organisers that reinforce key concepts. We provide revision guides, work-book and revision packs with a variety of extra revision resources throughout to continue to assess, improve and consolidate learning.

CURRICULUM AND ASSESSMENT PLAN

Alevel CS YEAR 13

ORMISTON SWB ACADEMY

INTENT

Our faculty mission is to equip students with a robust understanding of computer science principles, practical programming skills, and the ability to apply computational thinking to solve real-world problems. We aim to develop students who are not only proficient users of technology but also creative thinkers and informed digital citizens. The curriculum is designed to challenge and **Develop Computational Thinking**: Teach students how to think logically, solve problems efficiently, and understand the principles of computer systems and networks, Advance P**rogramming Proficiency**: Introduce students to programming languages such as Python, ensuring they understand basic syntax, control structures, and algorithms. Mature **Digital Literacy skills**: Ensure students are competent and safe users of technology, understanding online safety, data privacy, and the ethical implications of technology.

Prior learning	GCSE *Computer Systems*, which explores hardware, networks, security, and ethical issues, and *Computational Thinking, Algorithms, and Programming*, which focuses on coding principles, logic, and problem-solving. Students develop analytical and programming skills, preparing for exams that assess both theoretical knowledge and practical application.
Conscious curriculum links	The **A-Level Computer Science curriculum** integrates key computational concepts with broader educational themes, fostering problem-solving, analytical thinking, and ethical considerations in technology. It covers programming, algorithms, data structures, cybersecurity, and the societal impacts of computing, ensuring students develop both technical expertise and an awareness of real-world applications
Extra-curricular	Bebras Computing Challenge is an international competition that promotes computational thinking among students aged 6 to 18. Organized in over 80 countries.

	AUTUMN 1	AUTUMN 2	SPRING 1	SPRING 2	SUMMER 1	SUMMER 2
TOPIC/KNOWLEDGE	**The characteristics of contemporary processors, input, output and storage devices** 1.1.3 Input/Output and storage 1.1.1 Structure and function of the processor 1.2.1 Operating systems **Types of software and the different methodologies used to develop software** 1.2.1 System software 1.2.2 Application generation 1,2.3 Software Development 1.2.4 Types of programming language **Unit 3 NEA** 3.2 Design the solution 3.3 Developing the solution	**How data is exchanged between different systems** 1.3.1 compression, encryption and hashing 1.3.2 Databases 1.3.3 Networking 1.3.4 web Technologies **The use of algorithms to describe problems and standard algorithms** 2.3.1 Algorithms **Unit 3 NEA** 3.3 Developing the solution 3.4 Evaluation	**How data is represented and stored within different structures, Different algorithms that cab b applied to these structures** 1.4.1 data types 1.4.2 Data structures 1.4.3 Boolean algebra **The use of algorithms to describe problems and standard algorithms** 2.3.1 Algorithms **2.2 Problem solving and programming** Selection Iteration Programming techniques Basic data structures Basic data structures Subroutines Object oriented programming	**Legal, Moral, Cultural and ethical issues** 1.5.1 Computer related legislation 1.5.2 Moral and ethical issues **Understand what is meant by computational Thinking** 2.1.1 Thinking abstractly 2.1.2 Thinking procedurally 2.1.4 Thinking logically 2.1.5 Thinking concurrently 2.2.2 Computational Methods **2.2 Problem solving and programming** Selection Iteration Programming techniques Basic data structures Basic data structures Subroutines Object oriented programming	**Exam Revision and preparation sessions.**	
SKILLS	A comprehensive list of all Alevel Computer Science skills can be found at: https://www.ocr.org.uk/images/170844-specification-accredited-a-level-gce-computer-science-h446.pdf					
ASSESSMENT	CS- 1.1 Contemporary Processors, I/O and Storage Devices & 2.3 Algorithms – Complex Data Structures Unit 3 – Programming Project CS – structure and function of a processor, ALU, control unit and registers CS – multicore and parallel systems CS – 1.2 Software & Software Development &	CS – 1.3 Exchanging Data CS – normalisation to £NF and SQL CS - Relational databases	2.3 Algorithms – Sorting and Searching CS – Object oriented languages CS – 1.4 Data Types, data structures and algorithms CS – Addition and subtraction of binary numbers including floating point	CS- 1.5 Legal, moral, cultural and ethical issues 2.1 Elements of computational thinking		
VOCAB	**Key terminology:** Hardware Secondary storage Optical Magnetic Solid state Storage Characteristic Estimate Overheads **Key terminology:** System Software, Operating System (OS), Virtual Machine, Utility Programs, Device Drivers, Multitasking, Time Slice, Processor Management, User Interface, Graphical User Interface, Command Line Interface, Peripheral Management, File Management, Utility Program, Operating System, Defragmentation, Compression, Lossy, Lossless, Encryption, Backup, Incremental, Full	**Key terminology:** • baud rate • bit rate • bandwidth • latency • protocol. • physical star topology • logical bus network topology	**Key terminology:** • Bit Nibble Byte Kilobyte Megabyte Gigabyte Terabyte Petabyte Binary Denary Conversion Transistor Bit Binary Denary• Mid Right Addition Subtraction Division modular division integer division negation AND OR NOT Greater than Less than Equal	**Key terminology:** Requirements Interaction Prototype Agile Abstraction Solution structure documented design iterative debugging feedback critical path testing errors boundary erroneous evaluation **Key terminology:** Ethical issues, Legal issues, Cultural issues, Environmental issues, Privacy issues, Stakeholder, Open source software, Proprietary software, Data Protection Act, Computer Misuse Act, Copyright Designs and Patents Act, Creative commons licensing, Freedom of Information Act		

READING SKILLS

Reading Skills

Pupils develop critical reading skills by interpreting and understanding complex programming documentation. Pupils learn to extract relevant information, understand technical jargon, and comprehend detailed instructions. Computing emphasizes the importance of clear and precise writing. Pupils engage in writing code comments, documentation, and project reports, which require them to explain their thought processes and solutions in a coherent manner. This practice enhances their ability to articulate complex ideas. Digital literacy skills pupils will understand how to navigate and utilise various software tools, platforms, and online resources which is crucial in today's digital world. This competence supports their ability to research, learn independently, and engage with digital content critically and safely.

CAREERS LINKS

CS Career Developments: Software Engineer, IT Support Specialist, Computer Systems Analyst, AI Engineer, Robotics Engineer. Developments: Software Engineer, Cyber Security Analyst, Cloud Solutions, Games developer, Robotics

CORE

In computing, these values support problem-solving, persistence in debugging, effective project management, and striving for high-quality work. The curriculum reinforces personal development, emotional wellbeing, and responsible online behaviour, helping pupils build resilience and strong character both digitally and personally.

SUPPORTING STUDENTS AT HOME

we support pupils' learning at home through a combination of structured homework tasks and knowledge organisers that reinforce key concepts. We provide revision guides, work-book and revision packs with a variety of extra revision resources throughout to continue to assess, improve and consolidate learning.