Curriculum Assessment Map: Year 10 Computer Science

	Autumn Term One	Autumn Term Two	Spring Term One	Spring Term Two	Summer Term One	Summer Term Two
Topic	**10.1 Algorithms & Programming 1**	**10.1 Algorithms & Programming 1 (Cont.)**	**10.2 Representing Data**	**10.3 – Systems Architecture**	**10.4 – Programming 2**	**10.5 – Extended Programming**
Key Learning & Skills	• Explain the principles of computational thinking. • Design, create and refine algorithms using pseudocode and flowcharts. • Complete trace tables to show the values of variables when an algorithm is executed. • Perform binary & linear search. • Perform Bubble, Merge & Insertion sort. .	• Compare the efficiency of sorting & searching algorithms. • Explain the purpose of & use variables, constants, operators, inputs, outputs and assignments in Python. • Explain the three constructs of programming used to control program flow. • Name and perform common comparison and arithmetic operations. • Explain the need for data types and their features (Integer, real, Boolean, character, string).	• Explain the purpose of binary numbers. • Name the units of measurement between bit and Petabyte. • Calculate data capacity requirements within a system. • Convert numbers between binary, denary and hexadecimal representation. • Explain the purpose and use of number systems. • Perform binary addition. • Explain the purpose of and perform binary shifts. • Explain the purpose of a character set. • Compare and contrast ASCII & Unicode • Explain how bitmap images are stored in a computer system. • Explain what metadata	• Explain the difference between primary memory and secondary storage. • Explain the differences between RAM and ROM. • Explain the purpose and operation of virtual memory. • Explain the characteristics of CPUs and their effect on performance (Clock speed, Cache size, Number of cores) • Explain the purpose of the CPU and the FDE cycle. • Explain the purpose and function of the ALU, CU, Cache and Registers • Explain features of Von-Neumann architecture devices.	• Perform string manipulation • Perform file handling in Python. • Create both 1D and 2D arrays to solve a given problem • Explain the purpose of subroutines, distinguishing between procedures and functions. • Explain how random numbers can be generated in Python. • Create subroutines in Python. • Explain why programs should be designed with defensive considerations and maintainability considerations. • Explain the purpose of validation methods and implement in Python. • Explain common conventions for improving program	• Explain what an algorithm is. • Students apply their prior learning from and 10.4 to complete an extended programming challenge. Students will: • Analyse a given problem. • Design a solution. • Develop the solution. • Test the solution. • Evaluate the solution.

Curriculum Assessment Map: Year 10 Computer Science

			is • Explain how the quality and size of image and sound files is affected by a range of factors. • Explain the need for compression • Explain the differences between lossy and lossless compression.	• Explain the purpose of embedded systems and name examples. • Create simple logic diagrams using AND, OR and NOT • Create truth tables for given logic diagrams. • Combine operators to create logic circuits.	maintainability. • Explain the purpose of testing and the various forms which exist. • Select and use suitable testing data. • Explain what SQL is • Explain how SQL can be used to interrogate data. • Build and execute SQL statements to interrogate data.	
End points	Written assessment	Written assessment	Written assessment	Written assessment	Written assessment	Written assessment
Informal (*formative*) Assessment	• Class/GRIT Tasks • Do Now Activities • Q&A • Class quiz	• Class/GRIT Tasks • Do Now Activities • Q&A • Class quiz	• Class/GRIT Tasks • Do Now Activities • Q&A • Class quiz	• Class/GRIT Tasks • Do Now Activities • Q&A • Class quiz	• Class/GRIT Tasks • Do Now Activities • Q&A • Class quiz	• Class/GRIT Tasks • Do Now Activities • Q&A • Class quiz
Formal (*summative*) Assessment	• End of unit assessment	• End of unit assessment	• End of unit assessment	• End of unit assessment	• End of unit assessment	• End of unit assessment • Mock Test

Curriculum encompassing literacy, careers and enrichment as well as interconnectivity with other subjects

Curriculum Assessment Map: Year 10 Computer Science

Key skills/content requirements at GCSE	
Theoretical Computing	**Programming**
Students will follow the OCR J277 Computer Science course. Students will develop an understanding of the following key areas within computer science: 1.1 – Systems architecture 1.2 – Memory and storage 1.3 – Computer networks, connections and protocols 1.4 – Network security 1.5 - Systems software 1.6 - Ethical, legal, cultural and environmental impacts of digital technology 2.1 – Algorithms 2.2 – Programming fundamentals 2.3 – Producing robust programs 2.4 – Boolean logic 2.5 – Programming languages and Integrated Development Environments It is essential that students have a strong grasp on the key vocabulary associated with the subject allowing them to describe and explain the theory behind how a computer works. Students must also have the knowledge of how to solve a problem and develop a solution. Students will need to understand a range of processes which occur within computers in order to explain how a computer ultimately responds to user input and guarantees security.	• Write algorithms to solve given problems. • Visually represent algorithms using flowcharts. • Write subroutines to make flowcharts more manageable. • Use Python to program solutions to a given problem. • Order code so that Python runs in sequence. • Store values in variables so they can be manipulated. • Take user input and store the values in variables. • Manipulate values using mathematical operators. • Use selection to decide what code to run based on a condition. • Make decisions using comparative operators. • Use iteration to repeat sections of code. • Work out when to use condition-controlled (while) or count-controlled (for) iteration. • Write functions to better organise programs. • Pass parameters between functions to make code more efficient. • Use inbuilt functions contained within libraries.

Curriculum Assessment Map: Year 10 Computer Science

Curriculum Vision

The computing department at Leasowes High School puts students at the heart of everything we do. We aim to focus on enthusing and engaging students with computer science courses which are on offer through the coverage of varied content at KS3 relating to the computing field. Our curriculum will initially focus on ensuring that students are digitally literate and able to use ICT across the curriculum before embarking upon our key focus of computer science, whereby students are stretched and challenged to use technology to create solutions to given problems.

Curriculum principles

Our curriculum will:

- Ensure that all students are equipped with ICT skills required to use technology across the curriculum and in the workplace.
- Ensure that students can use technology in a safe and respectful manner, recognising the difference between safe and unsafe activities and behaviours online, and understanding how to respond to events that make them feel unsafe or uncomfortable.
- Equip students with the understanding of how computers work, giving them the skills to perform basic fixes to their own devices.
- Develop an understanding of cyber security including the risks, prevention, and remedy methods available amongst our students.
- Develop problem solving skills (abstraction, decomposition, sequence, selection, iteration) which are applicable to a range of other subjects through the solving of problems which are encountered in the computing industry.
- Focus on developing resilient learners who are not ashamed to make mistakes; we only learn from them.
- Introduce students to multiple programming languages to enable students to analyse, design, develop, test and evaluate solutions to problems.
- Be responsive to changes in technology, ensuring that content is reviewed on a regular basis and is relevant to technology being used in society.
- Provide extra-curricular opportunities to engage with computing beyond the classroom through after school clubs, trips and visitors which are relevant to the computing field.
- Explore a range of Ethical, Environmental, Cultural, Privacy and Legal issues related to the development of technology.

Theoretical Computing – Throughout study students will develop their understanding of the technology used to power the modern-day systems which we rely on. Developing understanding of networking, security, inner workings, and the storage of data. In addition to this we provide extensive understanding surrounding the origins of computing and a breadth of history.

Programming – Students will develop their ability to recognise a problem, extract important information, design a solution to a problem, implement using code, test and evaluate fully. Students will be introduced to concepts including variables, sequence, arithmetic operators, Boolean operators, selection, iteration, file handling, regular expressions, time handling, random numbers, subroutines and functions.

Literacy - Deliberate practice of keywords proves invaluable in enhancing literacy skills. By highlighting specific vocabulary and terminology, educators can enrich students' understanding and communication abilities. Actively encouraging students to engage with and employ these keywords in their work facilitates vocabulary expansion and cultivates a deeper comprehension of the subject matter. Mastery of key terms empowers students to express their ideas effectively and analyse academic resources within their respective fields. Ultimately, deliberate practice of keywords propels students towards excellence in their unit of work.

Curriculum Assessment Map: Year 10 Computer Science

SMSC in computer science

Spiritual development in computer science
Students are continually reflecting on their own lives and the lives of others as they look at various Computing case studies. Students debate and formulate their own set of values and beliefs through case studies as they share their own experiences. Computing is an area of rapid development and change; this provides students with the opportunity to reflect upon this progress and potential new technologies which will be developed in time.

Moral development in computer science
Within computing, it is important to consider many areas of the human impact technology has. Society is not only becoming more reliant on technology, but the increasing rate in which computers are updated causes substantial waste, as well as increased carbon footprint in line with their increased production. Students will investigate the use of social-networking and cyber bullying, whilst learning about the legal implications of immoral acts undertaken online. Students will consider where boundaries should lie and the impact of computing on the environment.

Social development in computer science
Computing can also help all students to express themselves clearly and to communicate. As students' progress through their learning, they will consider more complex social needs and are encouraged to research and work to find appropriate solutions to issues that may affect particular groups within society.

Cultural development in computer science
With the increased use of social media sites, people are becoming more culturally aware due to the diversity of content posted online for all of the world to see. Computational thinking encourages problem solving and thinking about how to solve an issue from another perspective – a valuable transferable skill that translates to many aspects of life. Students will consider the positive and negative effects of computing upon various groups of people.

Curriculum Assessment Map: Year 11 Computer Science

	Autumn Term One	Autumn Term Two	Spring Term One	Spring Term Two	Summer Term One	Summer Term Two
Topic	**11.1 – Networking & Security**	**11.2 – Software**	**11.2 – Software (cont.)**	**11.3 – The Impact of Computing**	**11.4 Revision Plan**	
Key Learning & Skills	• . Explain the characteristics of LAN and WAN networks. • Explain factors which affect the performance of a network. • The roles of computers in CS and P2P networks. • Explain the role of hardware required to create a LAN network. • Explain the concepts of DNS, Hosting, the cloud and webservers. • Compare star and mesh topologies. • Name modes of connection and explain their use within networks. • Explain the purpose of addressing. • Name common networking standards • Explain the purpose of common network protocols. • Explain the concept of layering. • Explain the various forms of threat to computer security. • Explain methods of identifying and preventing network vulnerabilities.	• Explain the purpose and functionality of operating systems (UI, Memory management, Multi-tasking, Peripheral management, Drivers, User management, File management) • Explain the difference between the purpose and function of applications and utility software. • Explain the role and purpose of encryption, defragmentation, and compression software. • Explain the purpose and features of high- and low-level languages. • Explain the purpose of translator software. • Explain the characteristics of compilers and interpreters. • Explain the tools and facilities which are available in an IDE. • Explain the difference between open- source and proprietary software licensing, and off-the-shelf and custom written software. • Recommend a type of	• What each function of an operating system does. • Features of a user interface • Memory management User management functions • File Management Purpose of utility software.	• Explain the impact of technology on wider society. • Evaluate the impact of computing in relation to ethical, legal, cultural, environmental and privacy related issued. • Explain the purpose of key legislation in the computer science field: • - Data Protection Act (2018) • - Computer Misuse Act (1990) • - Copyright, Designs and Patents Act (1988)	• Topics to be determined as appropriate. This will be guided by the mock assessments and departmental data collected throughout the duration of the course.	

Curriculum Assessment Map: Year 11 Computer Science

		license for a given scenario.				
End points	Written assessment	Written assessment	Written assessment			
Informal (*formative*) Assessment	• Class/GRIT Tasks • Do Now Activities • Q&A • Class quiz	• Class/GRIT Tasks • Do Now Activities • Q&A • Class quiz	• Class/GRIT Tasks • Do Now Activities • Q&A • Class quiz	• Class/GRIT Tasks • Do Now Activities • Q&A • Class quiz	• Class/GRIT Tasks • Do Now Activities • Q&A • Class quiz	
Formal (*summative*) Assessment	• End of unit assessment	• End of unit assessment • Mock Test	• End of unit assessment	• Practise exam papers • Mock Test	• Real Exam	

Curriculum encompassing literacy, careers and enrichment as well as interconnectivity with other subjects

Curriculum Assessment Map: Year 11 Computer Science

Key skills/content requirements at GCSE

Theoretical Computing

Students will follow the OCR J277 Computer Science course. Students will develop an understanding of the following key areas within computer science:

1.1 – Systems architecture
1.2 – Memory and storage
1.3 – Computer networks, connections and protocols
1.4 – Network security
1.5 - Systems software
1.6 - Ethical, legal, cultural and environmental impacts of digital technology
2.1 – Algorithms
2.2 – Programming fundamentals
2.3 – Producing robust programs
2.4 – Boolean logic
2.5 – Programming languages and Integrated Development Environments

It is essential that students have a strong grasp on the key vocabulary associated with the subject allowing them to describe and explain the theory behind how a computer works. Students must also have the knowledge of how to solve a problem and develop a solution.

Students will need to understand a range of processes which occur within computers in order to explain how a computer ultimately responds to user input and guarantees security.

Programming

- Write algorithms to solve given problems.
- Visually represent algorithms using flowcharts.
- Write subroutines to make flowcharts more manageable.
- Use Python to program solutions to a given problem.
- Order code so that Python runs in sequence.
- Store values in variables so they can be manipulated.
- Take user input and store the values in variables.
- Manipulate values using mathematical operators.
- Use selection to decide what code to run based on a condition.
- Make decisions using comparative operators.
- Use iteration to repeat sections of code.
- Work out when to use condition-controlled (while) or count-controlled (for) iteration.
- Write functions to better organise programs.
- Pass parameters between functions to make code more efficient.
- Use inbuilt functions contained within libraries.

Curriculum Assessment Map: Year 11 Computer Science

Curriculum Vision

The computing department at Leasowes High School puts students at the heart of everything we do. We aim to focus on enthusing and engaging students with computer science courses which are on offer through the coverage of varied content at KS3 relating to the computing field. Our curriculum will initially focus on ensuring that students are digitally literate and able to use ICT across the curriculum before embarking upon our key focus of computer science, whereby students are stretched and challenged to use technology to create solutions to given problems.

Curriculum principles

Our curriculum will:

- Ensure that all students are equipped with ICT skills required to use technology across the curriculum and in the workplace.
- Ensure that students can use technology in a safe and respectful manner, recognising the difference between safe and unsafe activities and behaviours online, and understanding how to respond to events that make them feel unsafe or uncomfortable.
- Equip students with the understanding of how computers work, giving them the skills to perform basic fixes to their own devices.
- Develop an understanding of cyber security including the risks, prevention, and remedy methods available amongst our students.
- Develop problem solving skills (abstraction, decomposition, sequence, selection, iteration) which are applicable to a range of other subjects through the solving of problems which are encountered in the computing industry.
- Focus on developing resilient learners who are not ashamed to make mistakes; we only learn from them.
- Introduce students to multiple programming languages to enable students to analyse, design, develop, test and evaluate solutions to problems.
- Be responsive to changes in technology, ensuring that content is reviewed on a regular basis and is relevant to technology being used in society.
- Provide extra-curricular opportunities to engage with computing beyond the classroom through after school clubs, trips and visitors which are relevant to the computing field.
- Explore a range of Ethical, Environmental, Cultural, Privacy and Legal issues related to the development of technology.

Theoretical Computing – Throughout study students will develop their understanding of the technology used to power the modern-day systems which we rely on. Developing understanding of networking, security, inner workings, and the storage of data. In addition to this we provide extensive understanding surrounding the origins of computing and a breadth of history.

Programming – Students will develop their ability to recognise a problem, extract important information, design a solution to a problem, implement using code, test and evaluate fully. Students will be introduced to concepts including variables, sequence, arithmetic operators, Boolean operators, selection, iteration, file handling, regular expressions, time handling, random numbers, subroutines and functions.

Literacy - Deliberate practice of keywords proves invaluable in enhancing literacy skills. By highlighting specific vocabulary and terminology, educators can enrich students' understanding and communication abilities. Actively encouraging students to engage with and employ these keywords in their work facilitates vocabulary expansion and cultivates a deeper comprehension of the subject matter. Mastery of key terms empowers students to express their ideas effectively and analyse academic resources within their respective fields. Ultimately, deliberate practice of keywords propels students towards excellence in their unit of work.

Curriculum Assessment Map: Year 11 Computer Science

SMSC in computer science

Spiritual development in computer science
Students are continually reflecting on their own lives and the lives of others as they look at various Computing case studies. Students debate and formulate their own set of values and beliefs through case studies as they share their own experiences. Computing is an area of rapid development and change; this provides students with the opportunity to reflect upon this progress and potential new technologies which will be developed in time.

Moral development in computer science
Within computing, it is important to consider many areas of the human impact technology has. Society is not only becoming more reliant on technology, but the increasing rate in which computers are updated causes substantial waste, as well as increased carbon footprint in line with their increased production. Students will investigate the use of social-networking and cyber bulling, whilst learning about the legal implications of immoral acts undertaken online. Students will consider where boundaries should lie and the impact of computing on the environment.

Social development in computer science
Computing can also help all students to express themselves clearly and to communicate. As students' progress through their learning, they will consider more complex social needs and are encouraged to research and work to find appropriate solutions to issues that may affect particular groups within society.

Cultural development in computer science
With the increased use of social media sites, people are becoming more culturally aware due to the diversity of content posted online for all of the world to see. Computational thinking encourages problem solving and thinking about how to solve an issue from another perspective – a valuable transferable skill that translates to many aspects of life. Students will consider the positive and negative effects of computing upon various groups of people.