Computer Science / Year 10 / 2022-2023

Year 10	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
Curriculum Content	Units from the specification covered: ➢ 1.1 Systems Architecture ➢ 1.2 Memory ➢ 1.3 Wired and Wireless Networks 1.1 System Architecture: **Component 1 –** Explain the purpose of Von Neumann Architecture. **Component 2** - Describe the function of the CPU within computers. **Component 3** – Explain the factors that impact performance of the CPU. **Component 4** – investigate the importance of embedded systems. 1.2 Memory: **Component 1** – Explain the difference between Main memory & secondary storage **Component 2** – Explain the importance of RAM **Component 3 –** Investigate the characteristics of Secondary storage **Component 4** – Explain the reason for the use of binary **Component 5** – Convert between the three	Units from the specification covered include: ➢ 1.3 Wired and Wireless Networks ➢ 1.4 System Security 1.3 Networks: **Component 1** – Describe the different types of computer network **Component 2** – Identify and describe the various pieces of network hardware **Component 3** – Explore the role of the internet **Component 4** – Describe the roles of network addressing **Component 5** - Explain how data is communicated between devices over a network 1.4 Network Security: **Component 1** – Identify the treats posed to devices and systems **Component 2** – Explain how the attacks work & their purpose **Component 3** – Explain how to limit the treats posed **Component 4** – Investigate the methods to remove vulnerabilities **Component 5** – Demonstrate between devices over a	Units from the specification covered include: ➢ 1.5 Systems Software ➢ 1.6 Ethical, Legal, Cultural and Environmental issues. 1.5 Operating System: **Component 1** – Describe the function of the OS **Component 2** – Explain the features of an OS (**MUMPS**) **Component 3** – Explain the purpose of utility software **Component 4** – Investigate the different types of utility software 1.6 The Issues in Computing (*Ethical, Legal, Cultural, Environmental & Privacy*) **Component 1** – Investigate how technology impacts the major issues surrounding Computing. **Component 2** – Discuss the legislation in place and the actions it prohibits **Component 3** – Discuss the pros & cons of open source software – comparing to proprietary software **Component 4** – recommend licences based of scenarios	Units from the specification covered include: ➢ Programming Project. ➢ Programming Techniques 2.2 Programming Techniques: **Component 1** – Construct programs utilising variables **Component 2** – Construct a program utilising user inputs **Component 3** – Construct programs using IF and nested IF statements. **Component 4** – Construct programs utilising conditional and count loops **Component 5**- Construct programs using 2D arrays **Component 6** – Utilise string manipulation within programs **Component 7** – Construct programs utilising text files to store results	Students to complete their programming project (20 hours), and will complete the following units: Students will complete 20 hours worth of independent programming during the course to satisfy the requirements of the exam board. 2.1 Algorithms – **Component 1** – Identify the need/purpose of algorithms **Component 2** – Explain the principles that are used to construct algorithms **Component 3** – Construct basic algorithms using either Exam Reference Language or Pseudocode. **Component 4** - Construct advanced programs utilising key elements from Spring 2 (*Component 4-7*)	Students to complete their NEA (20 hours), and will complete the following units: ➢ 2.3 Producing Robust Programs – **Component 1** – Identify the issues a programmer should consider to ensure that a program caters for all likely input values **Component 2** - Explain how to deal with invalid data in a program **Component 3** - Explain why commenting is useful and apply this Appropriately. **Component 4** – Investigate the different types of testing used **Component 5** – To investigate the types of tests used to thoroughly test a program ➢ 2.4 Computational Logic – **Component 1** – create truth tables based off any given scenario. **Component 2** - Identify the different logic gates **Component 3** – Construct Logic diagrams based off scenarios given ➢ 2.5 Translators and facilities of languages – **Component 1** – discuss the differences between High & Low level languages

	systems (Denary, Binary, Hex) **Component 6** – Explain how Images are represented within a computer **Component 7** - Explain how sound is represented within a computer **Component 8** – Explain the use of character sets	network and how these limit the attacks				**Component 2** - Investigate the need for translators and their functions **Component 3** – Investigate the purpose of an IDE **Component 4** – Discuss the uses that an IDE provides Students will complete 20 hours worth of independent programming during the course to satisfy the requirements of the exam board.
Prior knowledge and skills (from previous year / key stage)	Students will have completed a unit based on the key components of a computer system (1.1, 1.2, 1.3) during KS3 and will be familiar with the basic parts of a computer system. Students will have completed 2 units of learning during KS3 based on network parts and types.	Students will have completed 2 units based on network components and types of networks at KS3. Students will have completed 3 units in KS3 where they have studied the key threats to a computer network, and how to prevent these threats.	Students will have very limited knowledge of the operating system as this is not focused on during KS3.	Students will have looked at programming in all 3 years of computing at KS3. Students in year 9 complete a basic python unit of work which provide a foundation and allow students to make connections between year 9 and year 10	Students will have looked at programming in all 3 years of computing at KS3. Students in year 9 complete a basic python unit of work which provide a foundation and allow students to make connections between year 9 and year 10	Students will have looked at programming in all 3 years of computing at KS3. Students in year 9 complete a basic python unit of work which provide a foundation and allow students to make connections between year 9 and year 10.
Assessment Objectives	AO1: Demonstrate knowledge and understanding of the key concepts and principles of Computer Science AO2: Apply knowledge and understanding of key concepts and principles of Computer Science.	AO1: Demonstrate knowledge and understanding of the key concepts and principles of Computer Science AO2: Apply knowledge and understanding of key concepts and principles of Computer Science.	AO1: Demonstrate knowledge and understanding of the key concepts and principles of Computer Science AO2: Apply knowledge and understanding of key concepts and principles of Computer Science.	AO1: Demonstrate knowledge and understanding of the key concepts and principles of Computer Science AO2: Apply knowledge and understanding of key concepts and principles of Computer Science. AO3: Analyse problems in computational terms: • to make reasoned judgements • to design, program, evaluate and refine solutions	AO1: Demonstrate knowledge and understanding of the key concepts and principles of Computer Science AO2: Apply knowledge and understanding of key concepts and principles of Computer Science. AO3: Analyse problems in computational terms: • to make reasoned judgements • to design, program, evaluate and refine solutions	AO1: Demonstrate knowledge and understanding of the key concepts and principles of Computer Science AO2: Apply knowledge and understanding of key concepts and principles of Computer Science. AO3: Analyse problems in computational terms: • to make reasoned judgements • to design, program, evaluate and refine solutions
Vocabulary / Key Subject Terminology	○ RAM ○ ROM ○ CPU ○ Memory ○ Process ○ Network ○ Cores ○ ALU ○ CU	○ Protocol ○ Layers ○ Wireless ○ Wired ○ Peer-to-peer ○ Client-server ○ Security ○ Transmission ○ Hub	○ Software ○ Operating ○ System ○ Interface ○ GUI ○ Command ○ Firmware ○ Management ○ Peripheral	○ Abstraction ○ Decomposition ○ Iteration ○ Selection ○ Sequence ○ Function ○ Procedure ○ Data types ○ Validation	○ Pseudocode ○ Flowchart ○ Abstraction ○ Decomposition ○ Iteration ○ Selection ○ Sequence ○ Function ○ Procedure	○ Pseudocode ○ Flowchart ○ Abstraction ○ Decomposition ○ Iteration ○ Selection ○ Sequence ○ Function ○ Procedure

	○ ROM ○ Storage ○ Binary ○ Hexadecimal ○ Conversion ○ Denary ○ Character set ○ ASCII ○ Unicode ○ Pixel ○ Resolution ○ Bit depth ○ Bit Rate ○ Metadata ○ Sample ○ Frequency ○ Analogue ○ Digital ○ Bitmap ○ Vector	○ Switch ○ Router ○ Ethernet ○ Malware ○ Virus ○ DOS ○ Injection ○ Penetration ○ Testing ○ Firewall ○ Domain ○ Server ○ Cloud ○ Virtual ○ Hardware	○ Encryption ○ Utility ○ Legislation ○ Software ○ Proprietary ○ Open Source ○ Data ○ Protection ○ Environmental ○ Cultural ○ Moral ○ Ethical ○ Misues ○ Copyright ○ Disposal ○	○ Maintainability ○ Error ○ Logical ○ Syntax ○ Iterative ○ Terminal ○ Comparative ○ Editor	○ Data types ○ Validation ○ Maintainability ○ Error ○ Logical ○ Syntax ○ Iterative ○ Terminal ○ Comparative ○ Editor	○ Data types ○ Validation ○ Maintainability ○ Error ○ Logical ○ Syntax ○ Iterative ○ Terminal ○ Comparative ○ Editor
Assessment 1	Mini assessments to be given every 2 weeks at least (see O: drive). Students should complete at least 2 mini assessments per unit for 1.1, 1.2 and 1.3 during this half-term.	Mini assessments to be given every 2 weeks at least (see O: drive). Students should complete at least 2 mini assessments per unit for 1.4, 1.5 for this half-term.	Mini assessments to be given every 2 weeks at least (see O: drive). Students should complete at least 2 mini assessments per unit for 1.6 and 1.7 for this half-term.	Mini assessments to be given every 2 weeks at least (see O: drive). Students should complete at least 2 mini assessments per unit for 1.7, 1.8 for this half-term.	Mini assessments to be given at least every 2 weeks (see O: drive). Students should complete any remaining tests from 1.7 or 1.8 that have been missed. Students will be completing their NEA for this topic and so a mini assessment for Paper 2 is not suitable.	Students will be completing their NEA for this topic and so a mini assessment for Paper 2 is not suitable.
Assessment 2	Students will complete a large assessment during the 2nd half-term based on taught content so far (1.1, 1.2 and 1.3).		Students will complete a full paper 1 assessment (2018 past paper) at the end of half-term 4. This paper will be out of 80.		Students will complete a full paper 2 assessment (2018 past paper) at the end of half-term 4. This paper will be capped to taught content (around 36 marks).	

Year 11	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
Curriculum Content	2.1 Algorithms – **Component 1** – Identify the need/purpose of algorithms **Component 2** – Explain the principles that are used to construct algorithms **Component 3** – Construct basic algorithms using either Exam Reference Language or Pseudocode. **Component 4** - Construct advanced programs utilising key elements from Spring 2 (*Component 4-7*) 2.2 Programming Techniques: **Component 1** – Construct programs utilising variables **Component 2** – Construct a program utilising user inputs **Component 3** – Construct programs using IF and nested IF statements. **Component 4** – Construct programs utilising conditional and count loops **Component 5**- Construct programs using 2D arrays **Component 6** – Utilise string manipulation within programs **Component 7** – Construct programs utilising text files to store results **REVISION LESSONS**	following units: ➢ 2.3 Producing Robust Programs – **Component 1** – Identify the issues a programmer should consider to ensure that a program caters for all likely input values **Component 2** - Explain how to deal with invalid data in a program **Component 3** - Explain why commenting is useful and apply this Appropriately. **Component 4** – Investigate the different types of testing used **Component 5** – To investigate the types of tests used to thoroughly test a program ➢ 2.4 Computational Logic – **Component 1** – create truth tables based off any given scenario. **Component 2** - Identify the different logic gates **Component 3** – Construct Logic diagrams based off scenarios given ➢ 2.5 Translators and facilities of languages – **Component 1** – discuss the differences between High & Low level languages **Component 2** - Investigate the need for translators and their functions	1.7 System Architecture: **Component 1 –** Explain the purpose of Von Neumann Architecture. **Component 2** - Describe the function of the CPU within computers. **Component 3** – Explain the factors that impact performance of the CPU. **Component 4** – investigate the importance of embedded systems. 1.8 Memory: **Component 1** – Explain the difference between Main memory & secondary storage **Component 2** – Explain the importance of RAM **Component 3 –** Investigate the characteristics of Secondary storage **Component 4** – Explain the reason for the use of binary **Component 5** – Convert between the three 1.9 Networks: **Component 1** – Describe the different types of computer network **Component 2** – Identify and describe the various pieces of network hardware **Component 3** – Explore the role of the internet **Component 4** – Describe the roles of network addressing	1.11 Operating System: **Component 1** – Describe the function of the OS **Component 2** – Explain the features of an OS (**MUMPS**) **Component 3** – Explain the purpose of utility software **Component 4** – Investigate the different types of utility software 1.12 The Issues in Computing (*Ethical, Legal, Cultural, Environmental & Privacy*) **Component 1** – Investigate how technology impacts the major issues surrounding Computing. **Component 2** – Discuss the legislation in place and the actions it prohibits **Component 3** – Discuss the pros & cons of open source software – comparing to proprietary software **Component 4** – recommend licences based of scenarios Students must also practice 8-12 mark questions that required extended writing techniques. The main topics of these questions are on paper 1, in particular 1.8. 2.1 Algorithms – **Component 4** - Construct advanced programs utilising key elements from Spring 2 (*Component 4-7*) **REVISION LESSONS**	Students will complete a series of revision lessons around key topics and question styles. Students must be given access to the past paper question format with the correct answers on the reverse for self and peer-assessment. Students will complete a range of topics from a variety of units with the aim being to complete tasks that are identified as weaknesses from the data collected through mini assessments and past paper mock exams. **REVISION LESSONS** Specifically targeted at the class, emphasis on common occurrences and weaknesses present during mastery tasks and Mock papers **Test walkthroughs will also be completed**	**N/A**

		Component 3 – Investigate the purpose of an IDE **Component 4** – Discuss the uses that an IDE provides **REVISION LESSONS**	**Component 5** - Explain how data is communicated between devices over a network 1.10 Network Security: **Component 1** – Identify the treats posed to devices and systems **Component 2** – Explain how the attacks work & their purpose **Component 3** – Explain how to limit the treats posed **Component 4** – Investigate the methods to remove vulnerabilities **Component 5** – Demonstrate between devices over a network and how these limit the attacks **REVISION LESSONS**			
Prior knowledge and skills (from previous year / key stage)	Students will have completed a unit based on the key components of a computer system (1.1, 1.2, 1.3) during KS3 and will be familiar with the basic parts of a computer system. Students will have completed 2 units of learning during KS3 based on network parts and types.	Students will have completed 2 units based on network components and types of networks at KS3. Students will have completed 3 units in KS3 where they have studied the key threats to a computer network, and how to prevent these threats.	Students will have very limited knowledge of the operating system as this is not focused on during KS3.	Students will have looked at programming in all 3 years of computing at KS3. Students in year 9 complete a basic python unit of work which provide a foundation and allow students to make connections between year 9 and year 10	Students will have looked at programming in all 3 years of computing at KS3. Students in year 9 complete a basic python unit of work which provide a foundation and allow students to make connections between year 9 and year 10	
Assessment Objectives	AO1: Demonstrate knowledge and understanding of the key concepts and principles of Computer Science AO2: Apply knowledge and understanding of key concepts and principles of Computer Science. AO3:	AO1: Demonstrate knowledge and understanding of the key concepts and principles of Computer Science AO2: Apply knowledge and understanding of key concepts and principles of Computer Science. AO3:	AO1: Demonstrate knowledge and understanding of the key concepts and principles of Computer Science AO2: Apply knowledge and understanding of key concepts and principles of Computer Science.	AO1: Demonstrate knowledge and understanding of the key concepts and principles of Computer Science AO2: Apply knowledge and understanding of key concepts and principles of Computer Science.	AO1: Demonstrate knowledge and understanding of the key concepts and principles of Computer Science AO2: Apply knowledge and understanding of key concepts and principles of Computer Science. AO3: Analyse problems in computational terms:	

	Analyse problems in computational terms: • to make reasoned judgements • to design, program, evaluate and refine solutions	Analyse problems in computational terms: • to make reasoned judgements • to design, program, evaluate and refine solutions			• to make reasoned judgements • to design, program, evaluate and refine solutions	
Vocabulary / Key Subject Terminology	○ Abstraction ○ Decomposition ○ Iteration ○ Selection ○ Sequence ○ Function ○ Procedure ○ Data types ○ Validation ○ Maintainability ○ Error ○ Logical ○ Syntax ○ Iterative ○ Terminal ○ Comparative ○ Editor ○	○ Pseudocode ○ Flowchart ○ Abstraction ○ Decomposition ○ Iteration ○ Selection ○ Sequence ○ Function ○ Procedure ○ Data types ○ Validation ○ Maintainability ○ Error ○ Logical ○ Syntax ○ Iterative ○ Terminal ○ Comparative ○ Editor	○ RAM ○ ROM ○ CPU ○ Memory ○ Process ○ Network ○ Cores ○ ALU ○ CU ○ ROM ○ Storage ○ Binary ○ Hexadecimal ○ Conversion ○ Denary ○ Character set ○ ASCII ○ Unicode ○ Pixel ○ Resolution ○ Bit depth ○ Bit Rate ○ Metadata ○ Sample ○ Frequency ○ Analogue ○ Digital ○ Bitmap ○ Vector	○ Software ○ Operating ○ System ○ Interface ○ GUI ○ Command ○ Firmware ○ Management ○ Peripheral ○ Encryption ○ Utility ○ Legislation ○ Software ○ Proprietary ○ Open Source ○ Data ○ Protection ○ Environmental ○ Cultural ○ Moral ○ Ethical ○ Misues ○ Copyright ○ Disposal	Specific to topic. Keywords can be found in the MTP's for each topic or look at CM's	
Assessment 1	Mini assessments to be given every 2 weeks at least (see O: drive). Students should complete at least 2 mini assessments per unit for2.1 & 2.2	Mini assessments to be given every 2 weeks at least (see O: drive). Students should complete at least 2 mini assessments per unit for 2.2, 2.4 & 2.5.	Mini assessments to be given every 2 weeks at least (see O: drive). Students should complete at least 2 mini assessments per unit for 1.1 & 1.2	Mini assessments to be given every 2 weeks at least (see O: drive). Students should complete at least 2 mini assessments per unit for 1.7, 1.8 for this half-term.	Mini assessments to be given at least every 2 weeks (see O: drive). Students should complete any remaining tests from 1.7 or 1.8 that have been missed. Students will be completing their NEA for this topic and so a mini assessment for Paper 2 is not suitable.	
Assessment 2	Students will complete a large assessment during the 2nd half-term based on taught content so far (1.1, 1.2 and 1.3).		Students will complete a full paper 1 assessment (2018 past paper) at the end of half-term 4. This paper will be out of 80.		Students will complete a full paper 2 assessment (2018 past paper) at the end of half-term 4. This paper will be capped to taught content (around 36 marks).	