Year 9: GCSE 9-1 AQA 8520 Computer Science

Autumn Term

Half Term 1

Unit/ Topic title: 3.1 Fundamentals of algorithms

Learning weeks: 5 weeks (1.5 lessons a week)

Key learning (knowledge and skills):
3.1 Fundamentals of algorithms

3.1.1 Representing algorithms

- Understand and explain the term algorithm.
- Understand and explain the term decomposition
- Understand and explain the term abstraction.
- Use a systematic approach to problem solving and algorithm creation representing those algorithms using pseudo-code and flowcharts.
- Explain simple algorithms in terms of their inputs, processing and outputs.
- Determine the purpose of simple algorithms.

Unit/ Topic title: 3.2 Programming
3.2.1 Data types
Understand the concept of a data type.
Understand and use the following appropriately: Integer, Real, Boolean, Character string.
3.2.2 Programming concepts
Use, understand and know how the following statement types can be combined in programs
Variable declaration constant declaration Assignment Iteration Selection and subroutine (procedure/function).
Use definite and indefinite iteration, including indefinite iteration with the condition(s) at the start or the end of the iterative structure
Use nested selection and nested iteration structures.
Use meaningful identifier names and know why it is important to use them
3.2.3 Arithmetic operations in a programming language

Spring Term

Half Term 1

Unit/ Topic title: 3.3 Fundamentals of data representation

Learning weeks: 5 weeks (1.5 lessons a week)
Key learning (knowledge and skills):

3.3.1 Number bases
Understand the following number bases:
• decimal (base 10)
• binary (base 2)
• hexadecimal (base 16).
Understand that computers use binary to represent all data and instructions.
Explain why hexadecimal is often used in computer science.

3.3.2 Converting between number bases
Understand how binary can be used to represent whole numbers.
Understand how hexadecimal can be used to represent whole numbers.
Be able to convert in both directions between:
• binary and decimal
• binary and hexadecimal
• decimal and hexadecimal.

3.3.3 Units of information
Know that:
• a bit is the fundamental unit of information
• a byte is a group of 8 bits.

Know that quantities of bytes can be described using prefixes.
Know the names, symbols and corresponding values for the decimal prefixes:
• kilo, 1 kB is 1,000 bytes
• mega, 1 MB is 1,000 kilobytes

Summer Term

Half Term 1

Unit/ Topic title: 3.4 Computer systems

Learning weeks: 5 weeks (1.5 lessons a week)
Key learning (knowledge and skills):

3.4.1 Hardware and software
Define the terms hardware and software and understand the relationship between them.

3.4.2 Boolean logic
Construct truth tables for the following logic gates: • NOT, • AND, • OR.
Construct truth tables for simple logic circuits.
Interpret the results of simple truth tables.
Create, modify and interpret simple logic circuit diagrams.

3.4.3 Software classification
Explain what is meant by (give examples) • system software • application software.
Understand the need for, and functions of, operating systems (OS) and utility programs.
Understand that the OS handles management of the: • processor(s), • memory, • I/O devices, • applications and • security.

3.4.4 Systems architecture
Explain the Von Neumann architecture.
Explain the role and operation of main memory and the following major components of a central processing unit (CPU): • arithmetic logic unit, • control unit, • clock, • bus.
Explain the effect of the following on the performance of the CPU:
• clock speed, • number of processor cores, • cache size, • cache type.
Understand and explain the Fetch-Execute cycle.

Be familiar with and be able to use: addition, subtraction, multiplication, real division, integer division, including remainders. **Assessment:** Fundamentals of Algorithms 1 Written questions paper: short-to-medium-answer questions **Key vocabulary:** Pseudocode, Flowcharts, Structure Charts, Sorting, Searching and Dry run testing using trace tables **Core texts:** • AQA GCSE (9-1) Computer Science by S Robson and P M Heathcote • AQA Computer Science for GCSE Student Book (AQA GCSE) by Steve Cushing • New GCSE Computer Science AQA Revision Guide • New GCSE Computer Science AQA Exam Practice Workbook • Learning to Program in Python 2017 by P M Heathcote **Key websites and media to support learning:** www.mrplahe.com, www.teach-ict.com, Dynamic learning	• giga, 1 GB is 1,000 Megabytes • tera, 1 TB is 1,000 Gigabytes. **3.3.4 Binary arithmetic** Be able to add together up to three binary numbers. Be able to apply a binary shift to a binary number. Describe situations where binary shifts can be used. **3.3.5 Character encoding** Understand what a character set is and be able to describe the following character encoding methods: • 7-bit ASCII • Unicode. Understand that character codes are commonly grouped and run in sequence within encoding tables. Describe the purpose of Unicode and the advantages of Unicode over ASCII. Know that Unicode uses the same codes as ASCII up to 127. **Assessment: Fundamentals of data representation**: Written questions paper: short-to-medium-answer questions **Key vocabulary:** Binary and hexadecimal, denary conversion, Character coding (ASCII & Unicode), Binary addition, Images, Sound and Compression (Huffman & RLE)	Understand the differences between main memory and secondary storage. Understand the differences between RAM and ROM. **Assessment:** Computer Systems 1 (Written questions paper: short-to-medium-answer questions) **Key vocabulary:** Hardware and software, Logic gates, Software classification, Application and system software, Operating system, System architecture, CPU, Memory, Secondary storage, Cloud.

Year 9: GCSE 9-1 AQA 8520 Computer Science

Autumn Term

Half Term 2

Unit/ Topic title: 3.1 Fundamentals of algorithms

Learning weeks: 7 weeks (1.5 lessons a week)

Key learning (knowledge and skills):

3.1.2 Efficiency of algorithms

- Understand that more than one algorithm can be used to solve the same problem.
- Compare the efficiency of algorithms explaining how some algorithms are more efficient than others in solving the same problem.

3.1.3 Searching algorithms

- Understand and explain how the linear search algorithm works.
- Understand and explain how the binary search algorithm works.
- Compare and contrast linear and binary search algorithm

3.1.4 Sorting algorithms

- Understand and explain how the merge sort algorithm works.
- Understand and explain how the bubble sort algorithm works.
- Compare and contrast merge sort and bubble sort algorithms.

Assessment: All topics in the Autumn term:
Fundamentals of Algorithms 2
Written questions paper: short-to-medium-answer questions

Key vocabulary: Pseudocode, Flowcharts, Structure Charts, Sorting, Searching and Dry run testing using trace tables

Spring Term

Half Term 2

Unit/ Topic title: 3.3 Fundamentals of data representation

Learning weeks: 5 weeks (1.5 lessons a week)

Key learning (knowledge and skills):

3.3.6 Representing images

Understand what a pixel is and be able to describe how pixels relate to an image and the way images are displayed.
Describe the following for bitmaps:
• size in pixels
• colour depth.
Know that the size of a bitmap image in pixels (width x height) is known as the image resolution.
Describe how a bitmap represents an image using pixels and colour depth.
Describe using examples how the number of pixels and colour depth can affect the file size of a bitmap image.
Calculate bitmap image file sizes based on the number of pixels and colour depth.
Convert binary data into a black and white image.
Convert a black and white image into binary data.

3.3.7 Representing sound

Understand that sound is analogue and that it must be converted to a digital form for storage and processing in a computer.
Understand that sound waves are sampled to create the digital version of sound.
Describe the digital representation of sound in terms of:
• sampling rate
• sample resolution.
Calculate sound file sizes based on the sampling rate and the sample resolution.

3.3.8 Data compression

Explain what data compression is.

Summer Term

Half Term 2

Unit/ Topic title: 3.4 Computer systems

Learning weeks: 5 weeks (1.5 lessons a week)

Key learning (knowledge and skills):

3.4.4 Systems architecture

Explain the Von Neumann architecture.
Explain the role and operation of main memory and the following major components of a central processing unit (CPU): • arithmetic logic unit, • control unit, • clock, • bus.
Explain the effect of the following on the performance of the CPU:
• clock speed, • number of processor cores, • cache size, • cache type.
Understand and explain the Fetch-Execute cycle.
Understand the differences between main memory and secondary storage. Understand the differences between RAM and ROM.

Assessment: GCSE End of year exam. Paper 1
Written questions paper: short-to-medium-answer and one long question(s)

Key vocabulary: **Key vocabulary:** Hardware and software, Logic gates, Software classification, Application and system software, Operating system, System architecture, CPU, Memory, Secondary storage, Cloud.

	Understand why data may be compressed and that there are different ways to compress data. Explain how data can be compressed using Huffman coding. Be able to interpret/create Huffman trees. Be able to calculate the number of bits required to store a piece of data compressed using Huffman coding. Be able to calculate the number of bits required to store a piece of uncompressed data in ASCII Explain how data can be compressed using run length encoding (RLE). Represent data in RLE frequency/data pairs. **Assessment: All topics in the Spring term:** **Fundamentals of data representation**: Written questions paper: short-to-medium-answer questions **Key vocabulary:** Binary and hexadecimal, denary conversion, Character coding (ASCII & Unicode), Binary addition, Images, Sound and Compression (Huffman & RLE)	

Year 9: GCSE 9-1 AQA 8520 Computer Science

Coursework: Python and using pseudo-code (Home Study, Intervention)

Autumn Term	Spring Term	Summer Term
Coursework: Python and using pseudo-code(Home Study, Intervention) **Unit/ Topic title: 3.2 Programming** **Learning weeks: 5 weeks: one lesson every fortnight** **Key learning (knowledge and skills):** **3.2.1 Data types** *Understand the concept of a data type.* Understand and use the following appropriately: Integer, Real, Boolean, Character string. **3.2.2 Programming concepts** Use, understand and know how the following statement types can be combined in programs Variable declaration constant declaration Assignment Iteration Selection and subroutine (procedure/function). Use definite and indefinite iteration, including indefinite iteration with the condition(s) at the start or the end of the iterative structure Use nested selection and nested iteration structures. Use meaningful identifier names and know why it is important to use them **3.2.3 Arithmetic operations in a programming language** Be familiar with and be able to use: addition, subtraction, multiplication, real division, integer division, including remainders. **Assessment:** (see Autumn Half – term 2) **Key vocabulary:** Data types**,** Variables and constants**,** Iteration**,** Selection**,** Arithmetic, relational and Boolean operators**,** Lists and 2D lists**,** Input/output file handing**,** String handling**,** Random number generator**,** Procedures and functions**,** Validation**,** Programming languages**,** Machine code**,** Assembler**,** High level language**,** Compiler, interpreters and assembler	**Coursework: Python and using pseudo-code** (Home Study, Intervention) **Unit/ Topic title: 3.2 Programming** **Learning weeks: 3 weeks: one lesson every fortnight.** Plus 2 lesson Practical test **Key learning (knowledge and skills):** **3.2.4 Relational operations in a programming language** Be familiar with and be able to use: equal to, not equal to, less than, greater than, less than or equal to, greater than or equal to. **3.2.5 Boolean operations in a programming language** Be familiar with and be able to use: NOT, AND and OR. **3.2.6 Data structures** Understand the concept of data structures Use arrays (or equivalent) in the design of solutions to simple problems. Use records (or equivalent) in the design of solutions to simple problems **3.2.7 Input/output and file handling** Be able to obtain user input from the keyboard. Be able to output data and information from a program to the computer display. Be able to read/write from/to a text file **Assessment:** Practical Programming test 1 – Design and Code and Test. (2 lessons) Coding test. (open book)	**Coursework: Python and using pseudo-code** (Home Study, Intervention) **Unit/ Topic title: 3.2 Programming** **Learning weeks: 4 weeks: one lesson every fortnight** **Key learning (knowledge and skills):** **3.2.8 String handling operations in a programming language** **Understand and be able to use:** Length, position, substring, concatenation, convert character to character code, convert character, code to character and string conversion operations. **3.2.9 Random number generation in a programming language** **Be able to use random number generation.** **3.2.10 Subroutines (procedures and functions)** Understand the concept of subroutines. Explain the advantages of using subroutines in programs Describe the use of parameters to pass data within programs. Use subroutines that return values to the calling routine. Know that subroutines may declare their own variables, called local variables, and that local variables usually: only exist while the subroutine is executing are only accessible within the subroutine. Use local variables and explain why it is good practice to do so **Assessment:** End of year exam paper – question on coding.

Year 10: GCSE 9-1 AQA 8520 Computer Science

Autumn Term

Half Term 1

Unit/ Topic title: 3.1 Fundamentals of algorithms 1

Learning weeks: 5 weeks (1.5 lessons a week)

Key learning (knowledge and skills):
3.1 Fundamentals of algorithms

3.1.1 Representing algorithms

- Understand and explain the term algorithm.
- Understand and explain the term decomposition
- Understand and explain the term abstraction.
- Use a systematic approach to problem solving and algorithm creation representing those algorithms using pseudo-code and flowcharts.
- Explain simple algorithms in terms of their inputs, processing and outputs.
- Determine the purpose of simple algorithms.

Unit/ Topic title: 3.2 Programming
3.2.1 Data types
Understand the concept of a data type.
Understand and use the following appropriately: Integer, Real, Boolean, Character string.
3.2.2 Programming concepts
Use, understand and know how the following statement types can be combined in programs
Variable declaration constant declaration Assignment Iteration Selection and subroutine (procedure/function).
Use definite and indefinite iteration, including indefinite iteration with the condition(s) at the start or the end of the iterative structure
Use nested selection and nested iteration structures.
Use meaningful identifier names and know why it is important to use them
3.2.3 Arithmetic operations in a programming language
Be familiar with and be able to use: addition, subtraction, multiplication, real division, integer division, including remainders.

Spring Term

Half Term 1

Unit/ Topic title: 3.3 Fundamentals of data representation

Learning weeks: 5 weeks (1.5 lessons a week)
Key learning (knowledge and skills):

3.3.1 Number bases
Understand the following number bases:
• decimal (base 10)
• binary (base 2)
• hexadecimal (base 16).
Understand that computers use binary to represent all data and instructions.
Explain why hexadecimal is often used in computer science.

3.3.2 Converting between number bases
Understand how binary can be used to represent whole numbers.
Understand how hexadecimal can be used to represent whole numbers.
Be able to convert in both directions between:
• binary and decimal
• binary and hexadecimal
• decimal and hexadecimal.

3.3.5 Character encoding

Understand what a character set is and be able to describe the following character encoding methods:
• 7-bit ASCII
• Unicode.

Understand that character codes are commonly grouped and run in sequence within encoding tables.

Describe the purpose of Unicode and the advantages of Unicode over ASCII.
Know that Unicode uses the same codes as ASCII up to 127.

Summer Term

Half Term 1

Unit/ Topic title: 3.4 Computer systems

Learning weeks: 5 weeks (1.5 lessons a week)
Key learning (knowledge and skills):

3.4.1 Hardware and software
Define the terms hardware and software and understand the relationship between them.

3.3.3 Units of information
Know that:
• a bit is the fundamental unit of information
• a byte is a group of 8 bits.

Know that quantities of bytes can be described using prefixes.
Know the names, symbols and corresponding values for the decimal prefixes:
• kilo, 1 kB is 1,000 bytes
• mega, 1 MB is 1,000 kilobytes
• giga, 1 GB is 1,000 Megabytes
• tera, 1 TB is 1,000 Gigabytes.

3.3.4 Binary arithmetic
Be able to add together up to three binary numbers.
Be able to apply a binary shift to a binary number.
Describe situations where binary shifts can be used.

3.4.3 Software classification
Explain what is meant by (give examples) • system software • application software.
Understand the need for, and functions of, operating systems (OS) and utility programs.
Understand that the OS handles management of the:
• processor(s), • memory, • I/O devices, • applications and • security.

3.4.4 Systems architecture
Explain the Von Neumann architecture.

Assessment: Fundamentals of Algorithms 1 Written questions paper: short-to-medium-answer questions **Key vocabulary:** Pseudocode, Flowcharts, Structure Charts, Sorting, Searching and Dry run testing using trace tables **Core texts:** • AQA GCSE (9-1) Computer Science by S Robson and P M Heathcote • AQA Computer Science for GCSE Student Book (AQA GCSE) by Steve Cushing • New GCSE Computer Science AQA Revision Guide • New GCSE Computer Science AQA Exam Practice Workbook • Learning to Program in Python 2017 by P M Heathcote **Key websites and media to support learning:** www.mrplahe.com, www.teach-ict.com, Dynamic learning	**3.4.2 Boolean logic** Construct truth tables for the following logic gates: • NOT, • AND, • OR. Construct truth tables for simple logic circuits. Interpret the results of simple truth tables. Create, modify and interpret simple logic circuit diagrams. **Assessment:** **Fundamentals of data representation**: Written questions paper: short-to-medium-answer questions **Key vocabulary:** Binary and hexadecimal, denary conversion, Character coding (ASCII & Unicode), Binary addition, Images, Sound and Compression (Huffman & RLE)	Explain the role and operation of main memory and the following major components of a central processing unit (CPU): • arithmetic logic unit, • control unit, • clock, • bus. Explain the effect of the following on the performance of the CPU: • clock speed, • number of processor cores, • cache size, • cache type. Understand and explain the Fetch-Execute cycle. Understand the differences between main memory and secondary storage. Understand the differences between RAM and ROM. **Assessment:** Computer Systems 1 (Written questions paper: short-to-medium-answer questions) **Key vocabulary:** Hardware and software, Logic gates, Software classification, Application and system software, Operating system, System architecture, CPU, Memory, Secondary storage, Cloud.

Year 11: GCSE 9-1 AQA 8520 Computer Science

Autumn Term	Spring Term	Summer Term
Half Term 1 **Unit/ Topic title**: 3.5 Fundamentals of computer networks **Learning weeks:** 5 weeks (2 lessons per week) **Key learning (knowledge and skills):** Define what a computer network is. Discuss the benefits and risks of computer networks. Describe the main types of computer network including: • Personal Area Network (PAN) • Local Area Network (LAN) and • Wide Area Network (WAN) Understand that networks can be wired or wireless.	**Half Term 1** **Unit/ Topic title: 3.6** Fundamentals of cyber security **Learning weeks:** 5 weeks (2 lessons per week) **Key learning (knowledge and skills):** Be able to define the term cyber security and be able to describe the main purposes of cyber security. **3.6.1 Cyber security threats** Understand and be able to explain the following cyber security threats: • social engineering techniques	**Half Term 1** **Unit/ Topic title:** 3.8 Aspects of software development **Learning weeks:** 5 weeks (2 lessons per week) **Key learning (knowledge and skills): Design** Be aware that before constructing a solution, the solution should be designed, for example planning data structures for the data model, designing algorithms, designing an appropriate modular structure for the solution and designing he user interface. **Implementation**

Discuss the benefits and risks of wireless networks as opposed to wired networks. Explain the following common network topologies: • star, • bus. Define the term 'network protocol'. **Assessment:** Computer Networks 1 Written questions paper: short-to-medium-answer questions **Key vocabulary:** PAN, LAN, WAN, Wired (cabling copper vs Fibre Optic) and Wireless networking , Network Topology – Bus and Star, Network Protocols (HTTP, HTTPS, FTP, UDP, SMTP, POP3, MAP, Ethernet Wi Fi, IP, TCP), TCPIP and the function of the each of the 4 layers	• malicious code • weak and default passwords • misconfigured access rights • removable media • unpatched and/or outdated software. Explain what penetration testing is and what it is used for. **3.6.1.1 Social engineering** Define the term social engineering. Describe what social engineering is and how it can be protected against. Explain the following forms of social engineering: • blagging (pretexting) • phishing • pharming • shouldering (or shoulder surfing). **3.6.1.2 Malicious code** Define the term 'malware'. Describe what malware is and how it can be protected against. Describe the following forms of malware: • computer virus • Trojan • spyware • adware. **3.6.2 Methods to detect and prevent cyber security threats** Understand and be able to explain the following security measures: • biometric measures (particularly for mobile devices) • password systems • CAPTCHA (or similar) • using email confirmations to confirm a user's identity • automatic software updates **Assessment: Fundamentals of cyber security**	Be aware that the models and algorithms need to be implemented in the form of data structures and code (instructions) that a computer can understand. **Testing** Be aware that the implementation must be tested for the presence of errors, using selected test data covering normal (typical), boundary (extreme) and erroneous data. **Evaluation/refining** Be aware that code created during implementation will often require refining as a result of testing. Be aware of the importance of assessing how well the solution meets the requirements of the problem and how the solution could be improved if the problem were to be revisited. **Assessment:** Coursework **Key vocabulary:** Design, Implementation, Testing and Evaluation

	Written questions paper: short-to-medium-answer questions and long extended questions. **Key vocabulary:** Threats, Basic of cybersecurity and preventative measures (e.g. Passwords, Biometric, Physical), Malicious software (Malware) and preventative measures, Social Engineering and Penetration testing	

7