Year 10	Year 11
Half Term 1	**Term 1**
Intro to Computer Science **J277/02: Computational thinking, algorithms and programming** Algorithms Programming fundamentals	**J277/02: Computational thinking, algorithms and programming** Algorithms Programming fundamentals
Half Term 2	**Term 2**
J277/02: Computational thinking, algorithms and programming Programming Fundamentals Producing robust programs Programming languages and integrated development environments	**J277/02: Computational thinking, algorithms and programming** Algorithms Programming fundamentals
Half Term 3	**Term 3**
J277/02: Computational thinking, algorithms and programming Boolean logic **J277/01: Computer System** Systems architecture Memory and storage	**J277/02: Computational thinking, algorithms and programming** Algorithms Programming fundamentals
Half Term 4	**Term 4**
J277/01: Computer System Computer networks, connections and protocols Network security	**J277/01: Computer System** Revision

Half Term 5	Term 5
J277/01: Computer System Systems software	**J277/01: Computer System** Revision
Half Term 6	**Term 6**
J277/01: Computer System Ethical, legal, cultural, and environmental impacts of digital technology	

Overview

KS4 Knowledge and Skills 2021-22 (Computer Science)

Year 10	Year 11
Term 1	**Term 1**

• Understand the significance of Computer Science within the modern world
• Learn about the expectations of Computer Science students
 	
• Understand Computational Thinking Principles:
  ○ Abstraction
  ○ Decomposition
  ○ Algorithmic thinking
 	
• Understand Designing, creating and refining algorithms:
  ○ Identify the inputs, processes, and outputs for a problem
  ○ Structure diagrams
  ○ Create, interpret, correct, complete, and refine algorithms using:
  ❖ Pseudocode
  ❖ Flowcharts
  ❖ Reference language/high-level programming language
  ○ Identify common errors
  ○ Trace tables
 	
• Understand Searching and Sorting Algorithms:
  ○ Binary Search
  ○ Linear Search
  ○ Bubble sort
  ○ Merge sort
  ○ Insertion sort
 	
• Understand Programming fundamentals:
  ○ The use of variables, constants, operators, inputs, outputs and assignments
  ○ The use of the three basic programming constructs used to control the flow of a program:
  ❖ Sequence
  ❖ Selection
	• Understand the significance of Computer Science within the modern world
	• Learn about the expectations of Computer Science students
	
	• Understand Computational Thinking Principles:
	  ○ Abstraction
	  ○ Decomposition
	  ○ Algorithmic thinking
	
	• Understand Designing, creating and refining algorithms:
	  ○ Identify the inputs, processes, and outputs for a problem
	  ○ Structure diagrams
	  ○ Create, interpret, correct, complete, and refine algorithms using:
	  ❖ Pseudocode
	  ❖ Flowcharts
	  ❖ Reference language/high-level programming language
	  ○ Identify common errors
	  ○ Trace tables
	
	• Understand Searching and Sorting Algorithms:
	  ○ Binary Search
	  ○ Linear Search
	  ○ Bubble sort
	  ○ Merge sort
	  ○ Insertion sort
	
	• Understand Programming fundamentals:
	  ○ The use of variables, constants, operators, inputs, outputs and assignments
	  ○ The use of the three basic programming constructs used to control the flow of a program:

❖ Iteration (count- and condition-controlled loops) o The common arithmetic operators o The common Boolean operators AND, OR and NOT ▪ Understand Data Types o Integer o Real o Boolean o Character and string o Casting	❖ Sequence ❖ Selection ❖ Iteration (count- and condition-controlled loops) o The common arithmetic operators o The common Boolean operators AND, OR and NOT ▪ Understand Data Types o Integer o Real o Boolean o Character and string o Casting
Term 2	**Term 2**
▪ Understand Additional Programming techniques: o The use of basic string manipulation o The use of basic file handling operations: ❖ Open ❖ Read ❖ Write ❖ Close o The use of records to store data o The use of SQL to search for data o The use of arrays (or equivalent) when solving problems, including both one-dimensional (1D) and two-dimensional arrays (2D) o How to use sub programs (functions and procedures) to produce structured code o Random number generation • Understand Defensive design considerations: o Anticipating misuse o Authentication o Input validation	▪ Understand Additional Programming techniques: o The use of basic string manipulation o The use of basic file handling operations: ❖ Open ❖ Read ❖ Write ❖ Close o The use of records to store data o The use of SQL to search for data o The use of arrays (or equivalent) when solving problems, including both one-dimensional (1D) and two-dimensional arrays (2D) o How to use sub programs (functions and procedures) to produce structured code o Random number generation • Understand Defensive design considerations: o Anticipating misuse o Authentication

○ Maintainability: ❖ Use of sub programs ❖ Naming conventions ❖ Indentation ❖ Commenting • Understand Testing ○ The purpose of testing ○ Types of testing: ❖ Iterative ❖ Final/terminal ○ Identify syntax and logic errors ○ Selecting and using suitable test data: ❖ Normal ❖ Boundary ❖ Invalid/Erroneous ○ Refining algorithms	○ Input validation ○ Maintainability: ❖ Use of sub programs ❖ Naming conventions ❖ Indentation ❖ Commenting • Understand Testing ○ The purpose of testing ○ Types of testing: ❖ Iterative ❖ Final/terminal ○ Identify syntax and logic errors ○ Selecting and using suitable test data: ❖ Normal ❖ Boundary ❖ Invalid/Erroneous ○ Refining algorithms
Term 3-4	**Term 3-4**
• Understand Boolean Logic: ○ Simple logic diagrams using the operators AND, OR and NOT ○ Truth tables ○ Combining Boolean operators using AND, OR and NOT ○ Applying logical operators in truth tables to solve problems • Understanding Programming languages and integrated development environments: ○ Characteristics and purpose of different levels of programming language: ❖ High-level languages o Low-level languages ❖ The purpose of translators	• Understand Boolean Logic: ○ Simple logic diagrams using the operators AND, OR and NOT ○ Truth tables ○ Combining Boolean operators using AND, OR and NOT ○ Applying logical operators in truth tables to solve problems • Understanding Programming languages and integrated development environments: ○ Characteristics and purpose of different levels of programming language: ❖ High-level languages o Low-level languages ❖ The purpose of translators

○ The characteristics of a compiler and an interpreter ○ Common tools and facilities available in an Integrated Development Environment (IDE): ❖ Editors ❖ Error diagnostics ❖ Run-time environment ❖ Translators • Understanding Systems Architecture ○ The purpose of the CPU: ❖ The fetch-execute cycle ○ Common CPU components and their function: ❖ ALU (Arithmetic Logic Unit) ❖ CU (Control Unit) ❖ Cache o Registers ○ Von Neumann architecture: ❖ MAR (Memory Address Register) ❖ MDR (Memory Data Register) ❖ Program Counter ❖ Accumulator • Understanding CPU performance: ○ How common characteristics of CPUs affect their performance: ❖ Clock speed ❖ Cache size ❖ Number of cores • Understanding Embedded Systems ○ The purpose and characteristics of embedded systems ○ Examples of embedded systems	○ The characteristics of a compiler and an interpreter ○ Common tools and facilities available in an Integrated Development Environment (IDE): ❖ Editors ❖ Error diagnostics ❖ Run-time environment ❖ Translators • Understanding Systems Architecture ○ The purpose of the CPU: ❖ The fetch-execute cycle ○ Common CPU components and their function: ❖ ALU (Arithmetic Logic Unit) ❖ CU (Control Unit) ❖ Cache o Registers ○ Von Neumann architecture: ❖ MAR (Memory Address Register) ❖ MDR (Memory Data Register) ❖ Program Counter ❖ Accumulator • Understanding CPU performance: ○ How common characteristics of CPUs affect their performance: ❖ Clock speed ❖ Cache size ❖ Number of cores • Understanding Embedded Systems ○ The purpose and characteristics of embedded systems ○ Examples of embedded systems

<table>
<tr>
<td>

- Understand Primary Storage:
  - The need for primary storage
  - The difference between RAM and ROM
  - The purpose of ROM in a computer system
  - The purpose of RAM in a computer system
  - Virtual memory

- Understand Secondary Storage:
  - The need for secondary storage
  - Common types of storage:
    - ❖ Optical
    - ❖ Magnetic
    - ❖ Solid state Suitable
  - Storage devices and storage media for a given application
  - The advantages and disadvantages of different storage devices and storage media relating to these characteristics:
    - ❖ Capacity
    - ❖ Speed
    - ❖ Portability
    - ❖ Durability
    - ❖ Reliability
    - ❖ Cost

- Understand Units:
  - The units of data storage:
    - ❖ Bit
    - ❖ Nibble (4 bits)
    - ❖ Byte (8 bits)
    - ❖ Kilobyte (1,000 bytes or 1 KB)
    - ❖ Megabyte (1,000 KB)
    - ❖ Gigabyte (1,000 MB)
    - ❖ Terabyte (1,000 GB)

</td>
<td>

- Understand Primary Storage:
  - The need for primary storage
  - The difference between RAM and ROM
  - The purpose of ROM in a computer system
  - The purpose of RAM in a computer system
  - Virtual memory

- Understand Secondary Storage:
  - The need for secondary storage
  - Common types of storage:
    - ❖ Optical
    - ❖ Magnetic
    - ❖ Solid state Suitable
  - Storage devices and storage media for a given application
  - The advantages and disadvantages of different storage devices and storage media relating to these characteristics:
    - ❖ Capacity
    - ❖ Speed
    - ❖ Portability
    - ❖ Durability
    - ❖ Reliability
    - ❖ Cost

- Understand Units:
  - The units of data storage:
    - ❖ Bit
    - ❖ Nibble (4 bits)
    - ❖ Byte (8 bits)
    - ❖ Kilobyte (1,000 bytes or 1 KB)
    - ❖ Megabyte (1,000 KB)
    - ❖ Gigabyte (1,000 MB)
    - ❖ Terabyte (1,000 GB)

</td>
</tr>
</table>

❖ Petabyte (1,000 TB) o How data needs to be converted into a binary format to be processed by a computer o Data capacity and calculation of data capacity requirements • Understand Data Storage: o Numbers: ❖ How to convert positive denary whole numbers to binary numbers (up to and including 8 bits) and vice versa ❖ How to add two binary integers together (up to and including 8 bits) and explain overflow errors which may occur ❖ How to convert positive denary whole numbers into 2-digit hexadecimal numbers and vice versa ❖ How to convert binary integers to their hexadecimal equivalents and vice versa ❖ Binary shifts o Characters ❖ The use of binary codes to represent characters ❖ The term 'character set' ❖ The relationship between the number of bits per character in a character set, and the number of characters which can be represented, e.g.: ➢ ASCII ➢ Unicode o Images: ❖ How an image is represented as a series of pixels, represented in binary ❖ Metadata ❖ The effect of colour depth and resolution on: ➢ The quality of the image	❖ Petabyte (1,000 TB) o How data needs to be converted into a binary format to be processed by a computer o Data capacity and calculation of data capacity requirements • Understand Data Storage: o Numbers: ❖ How to convert positive denary whole numbers to binary numbers (up to and including 8 bits) and vice versa ❖ How to add two binary integers together (up to and including 8 bits) and explain overflow errors which may occur ❖ How to convert positive denary whole numbers into 2-digit hexadecimal numbers and vice versa ❖ How to convert binary integers to their hexadecimal equivalents and vice versa ❖ Binary shifts o Characters ❖ The use of binary codes to represent characters ❖ The term 'character set' ❖ The relationship between the number of bits per character in a character set, and the number of characters which can be represented, e.g.: ➢ ASCII ➢ Unicode o Images: ❖ How an image is represented as a series of pixels, represented in binary ❖ Metadata ❖ The effect of colour depth and resolution on:

➢ The size of an image file ○ Sound: ❖ How sound can be sampled and stored in digital form ❖ The effect of sample rate, duration and bit depth on: ➢ The playback quality ➢ The size of a sound file • Understand Compression: ○ The need for compression ○ Types of compression: ❖ Lossy ❖ Lossless • Understand Network Topologies: ○ Types of network: ❖ LAN (Local Area Network) ❖ WAN (Wide Area Network) ○ Factors that affect the performance of networks ○ The different roles of computers in a client-server and a peer-to peer network ○ The hardware needed to connect stand-alone computers into a Local Area Network: ❖ Wireless access points ❖ Routers ❖ Switches ❖ NIC (Network Interface Controller/Card) ❖ Transmission media ○ The Internet as a worldwide collection of computer networks: ❖ DNS (Domain Name Server) ❖ Hosting ❖ The Cloud	➢ The quality of the image ➢ The size of an image file ○ Sound: ❖ How sound can be sampled and stored in digital form ❖ The effect of sample rate, duration and bit depth on: ➢ The playback quality ➢ The size of a sound file • Understand Compression: ○ The need for compression ○ Types of compression: ❖ Lossy ❖ Lossless • Understand Network Topologies: ○ Types of network: ❖ LAN (Local Area Network) ❖ WAN (Wide Area Network) ○ Factors that affect the performance of networks ○ The different roles of computers in a client-server and a peer-to peer network ○ The hardware needed to connect stand-alone computers into a Local Area Network: ❖ Wireless access points ❖ Routers ❖ Switches ❖ NIC (Network Interface Controller/Card) ❖ Transmission media ○ The Internet as a worldwide collection of computer networks: ❖ DNS (Domain Name Server) ❖ Hosting

❖ Web servers and clients o Star and Mesh network topologies • Understanding wire and wireless networks, protocols and layers o Modes of connection: ❖ Wired ➢ Ethernet ❖ Wireless ➢ Wi-Fi ➢ Bluetooth o Encryption o IP addressing and MAC addressing o Standards o Common protocols including: ❖ TCP/IP (Transmission Control Protocol/Internet Protocol) ❖ HTTP (Hyper Text Transfer Protocol) ❖ HTTPS (Hyper Text Transfer Protocol Secure) ❖ FTP (File Transfer Protocol) ❖ POP (Post Office Protocol) ❖ IMAP (Internet Message Access Protocol) ❖ SMTP (Simple Mail Transfer Protocol) o The concept of layers • Understanding network security: o Threats to computer systems and networks: ❖ Forms of attack: ➢ Malware ➢ Social engineering, e.g. phishing, people as the 'weak point' ➢ Brute-force attacks	❖ The Cloud ❖ Web servers and clients o Star and Mesh network topologies • Understanding wire and wireless networks, protocols and layers o Modes of connection: ❖ Wired ➢ Ethernet ❖ Wireless ➢ Wi-Fi ➢ Bluetooth o Encryption o IP addressing and MAC addressing o Standards o Common protocols including: ❖ TCP/IP (Transmission Control Protocol/Internet Protocol) ❖ HTTP (Hyper Text Transfer Protocol) ❖ HTTPS (Hyper Text Transfer Protocol Secure) ❖ FTP (File Transfer Protocol) ❖ POP (Post Office Protocol) ❖ IMAP (Internet Message Access Protocol) ❖ SMTP (Simple Mail Transfer Protocol) o The concept of layers • Understanding network security: o Threats to computer systems and networks: ❖ Forms of attack: ➢ Malware

➢ Denial of service attacks ➢ Data interception and theft ➢ The concept of SQL injection ○ Identifying and preventing vulnerabilities: ❖ Common prevention methods: ➢ Penetration testing ➢ Anti-malware software ➢ Firewalls ➢ User access levels ➢ Passwords ➢ Encryption ➢ Physical security	➢ Social engineering, e.g. phishing, people as the 'weak point' ➢ Brute-force attacks ➢ Denial of service attacks ➢ Data interception and theft ➢ The concept of SQL injection ○ Identifying and preventing vulnerabilities: ❖ Common prevention methods: ➢ Penetration testing ➢ Anti-malware software ➢ Firewalls ➢ User access levels ➢ Passwords ➢ Encryption ➢ Physical security
Term 5-6	**Term 5-6**
• Understanding System Software: ○ Operating Systems: ❖ The purpose and functionality of operating systems: ➢ User interface ➢ Memory management and multitasking ➢ Peripheral management and drivers ➢ User management ➢ File management ○ Utility Software ❖ The purpose and functionality of utility software ❖ Utility system software: ➢ Encryption software ➢ Defragmentation ➢ Data compression	• Understanding System Software: ○ Operating Systems: ❖ The purpose and functionality of operating systems: ➢ User interface ➢ Memory management and multitasking ➢ Peripheral management and drivers ➢ User management ➢ File management ○ Utility Software ❖ The purpose and functionality of utility software ❖ Utility system software: ➢ Encryption software ➢ Defragmentation ➢ Data compression

- Understanding Ethical, legal, cultural, environmental impacts of digital technology:
  - Impacts of digital technology on wider society including:
    - ❖ Ethical issues
    - ❖ Legal issues
    - ❖ Cultural issues
    - ❖ Environmental issues
    - ❖ Privacy issues
  - Legislation relevant to Computer Science:
    - ❖ The Data Protection Act 2018
    - ❖ Computer Misuse Act 1990
    - ❖ Copyright Designs and Patents Act 1988
    - ❖ Software licences (i.e. open source and proprietary)

- Understanding Ethical, legal, cultural, environmental impacts of digital technology:
  - Impacts of digital technology on wider society including:
    - ❖ Ethical issues
    - ❖ Legal issues
    - ❖ Cultural issues
    - ❖ Environmental issues
    - ❖ Privacy issues
  - Legislation relevant to Computer Science:
    - ❖ The Data Protection Act 2018
    - ❖ Computer Misuse Act 1990
    - ❖ Copyright Designs and Patents Act 1988
    - ❖ Software licences (i.e. open source and proprietary)