| Year 10 | Year 11 |
|---|---|
| **Term 1** | **Term 1** |
| Intro to Computer Science<br><br>**J277/02: Computational thinking, algorithms and programming**<br>Algorithms<br>Programming fundamentals | **J277/02: Computational thinking, algorithms and programming**<br>Algorithms<br>Programming fundamentals |
| **Term 2** | **Term 2** |
| **J277/02: Computational thinking, algorithms and programming**<br>Programming Fundamentals<br>Producing robust programs<br>Programming languages and integrated development environments | **J277/02: Computational thinking, algorithms and programming**<br>Programming Fundamentals<br>Producing robust programs<br>Programming languages and integrated development environments |
| **Term 3** | **Term 3** |
| **J277/02: Computational thinking, algorithms and programming**<br>Boolean logic<br><br>**J277/01: Computer System**<br>Systems architecture<br>Memory and storage | **J277/02: Computational thinking, algorithms and programming**<br>Boolean logic<br><br>**J277/01: Computer System**<br>Systems architecture<br>Memory and storage |
| **Term 4** | **Term 4** |
| **J277/01: Computer System**<br>Computer networks, connections and protocols<br>Network security | **J277/01: Computer System**<br>Computer networks, connections and protocols<br>Network security |
| **Term 5** | **Term 5** |
| **J277/01: Computer System**<br>Systems software | **J277/01: Computer System**<br>Systems software |
| **Term 6** | **Term 6** |

| Year 10 | Year 11 |
|---|---|
| **Term 1** | **Term 1** |
| • Understand the significance of Computer Science within the modern world<br>• Learn about the expectations of Computer Science students<br><br>• Understand Computational Thinking Principles:<br>  ○ Abstraction<br>  ○ Decomposition<br>  ○ Algorithmic thinking<br><br>• Understand Designing, creating and refining algorithms:<br>  ○ Identify the inputs, processes, and outputs for a problem<br>  ○ Structure diagrams<br>  ○ Create, interpret, correct, complete, and refine algorithms using:<br>  ❖ Pseudocode<br>  ❖ Flowcharts<br>  ❖ Reference language/high-level programming language<br>  ○ Identify common errors<br>  ○ Trace tables<br><br>• Understand Searching and Sorting Algorithms:<br>  ○ Binary Search<br>  ○ Linear Search<br>  ○ Bubble sort<br>  ○ Merge sort<br>  ○ Insertion sort<br><br>• Understand Programming fundamentals:<br>  ○ The use of variables, constants, operators, inputs, outputs and assignments<br>  ○ The use of the three basic programming constructs used to control the flow of a program:<br>  ❖ Sequence<br>  ❖ Selection | • Understand the significance of Computer Science within the modern world<br>• Learn about the expectations of Computer Science students<br><br>• Understand Computational Thinking Principles:<br>  ○ Abstraction<br>  ○ Decomposition<br>  ○ Algorithmic thinking<br><br>• Understand Designing, creating and refining algorithms:<br>  ○ Identify the inputs, processes, and outputs for a problem<br>  ○ Structure diagrams<br>  ○ Create, interpret, correct, complete, and refine algorithms using:<br>  ❖ Pseudocode<br>  ❖ Flowcharts<br>  ❖ Reference language/high-level programming language<br>  ○ Identify common errors<br>  ○ Trace tables<br><br>• Understand Searching and Sorting Algorithms:<br>  ○ Binary Search<br>  ○ Linear Search<br>  ○ Bubble sort<br>  ○ Merge sort<br>  ○ Insertion sort<br><br>• Understand Programming fundamentals:<br>  ○ The use of variables, constants, operators, inputs, outputs and assignments<br>  ○ The use of the three basic programming constructs used to control the flow of a program: |

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

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

| | |
|---|---|
| • Understand Primary Storage:<br>  ○ The need for primary storage<br>  ○ The difference between RAM and ROM<br>  ○ The purpose of ROM in a computer system<br>  ○ The purpose of RAM in a computer system<br>  ○ Virtual memory<br><br>• Understand Secondary Storage:<br>  ○ The need for secondary storage<br>  ○ Common types of storage:<br>  ❖ Optical<br>  ❖ Magnetic<br>  ❖ Solid state Suitable<br><br>  ○ Storage devices and storage media for a given application<br>  ○ The advantages and disadvantages of different storage devices and storage media relating to these characteristics:<br>  ❖ Capacity<br>  ❖ Speed<br>  ❖ Portability<br>  ❖ Durability<br>  ❖ Reliability<br>  ❖ Cost<br><br>• Understand Units:<br>  ○ The units of data storage:<br>  ❖ Bit<br>  ❖ Nibble (4 bits)<br>  ❖ Byte (8 bits)<br>  ❖ Kilobyte (1,000 bytes or 1 KB)<br>  ❖ Megabyte (1,000 KB)<br>  ❖ Gigabyte (1,000 MB)<br>  ❖ Terabyte (1,000 GB) | • Understand Primary Storage:<br>  ○ The need for primary storage<br>  ○ The difference between RAM and ROM<br>  ○ The purpose of ROM in a computer system<br>  ○ The purpose of RAM in a computer system<br>  ○ Virtual memory<br><br>• Understand Secondary Storage:<br>  ○ The need for secondary storage<br>  ○ Common types of storage:<br>  ❖ Optical<br>  ❖ Magnetic<br>  ❖ Solid state Suitable<br><br>  ○ Storage devices and storage media for a given application<br>  ○ The advantages and disadvantages of different storage devices and storage media relating to these characteristics:<br>  ❖ Capacity<br>  ❖ Speed<br>  ❖ Portability<br>  ❖ Durability<br>  ❖ Reliability<br>  ❖ Cost<br><br>• Understand Units:<br>  ○ The units of data storage:<br>  ❖ Bit<br>  ❖ Nibble (4 bits)<br>  ❖ Byte (8 bits)<br>  ❖ Kilobyte (1,000 bytes or 1 KB)<br>  ❖ Megabyte (1,000 KB)<br>  ❖ Gigabyte (1,000 MB)<br>  ❖ Terabyte (1,000 GB) |

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

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

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

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