

What are the aims and intentions of this curriculum?

The aim of our Key Stage 4 Curriculum is to:

- Understand and apply the fundamental principles and concepts of data representation.
- Think creatively, innovatively, analytically, logically and critically in developing robust programs.
- Analyse problems in computational terms through practical experience of solving such problems, including designing, writing and debugging programs.

Term	Topics	Knowledge and key terms	Skills developed	Assessment
Autumn 1	Data Representation	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 (for example ASCII, extended ASCII and Unicode). Images • how an image is represented as a series of pixels represented in binary • metadata included in the file • 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 • how sampling intervals and other factors affect the size of a sound file and the quality of its playback: ✓ sample size ✓ bit rate ✓ sampling frequency. Compression • need for compression • types of compression: ✓ lossy ✓ lossless	• Explain how characters are represented in binary. • Calculate the ASCII code for any character. • Calculate the size of a text file. • Explain how images are represented in binary. • Calculate the size of an image file. • Explain how sound is represented in binary. • Calculate the size of an audio file. • Explain the disadvantages of large image and audio files. • Explain how file compression reduces the size of files. • Explain the differences between lossless and lossy file compression.	• Group Presentations • Seneca Online activities • Case Studies • End of topic quiz • End of term test • Microsoft Teams collaborative activities. • Mock Exams • Home work • Class Discussions • Topic Worksheets • Past Paper question sheets

Autumn 2

Producing Robust Programs

&

Programming Project

- defensive design considerations:
 - ✓ input sanitisation/validation
 - ✓ planning for contingencies
 - ✓ anticipating misuse
 - ✓ authentication
- maintainability:
 - ✓ comments
 - ✓ indentation
- the purpose of testing
- types of testing:
 - ✓ iterative
 - ✓ final/terminal
- how to identify syntax and logic errors
- selecting and using suitable test data

- Explain why user input is needed.
- Describe ways in which data input can be validated.
- Follow the systems development cycle to analyse problems, design and implement solutions and test the outcomes.

- Individual Presentations
- Seneca Online activities
- Case Studies
- End of topic quiz
- End of term test
- Microsoft Teams collaborative activities.
- Programming Practical
- Mock Exams
- Home work
- Class Discussions
- Electronic and hardcopy Portfolio
- Topic Worksheets
- Past Paper question sheets

Spring 1

Computational Logic

&

Programming Project

- why data is represented in computer systems in binary form
- simple logic diagrams using the operations AND, OR and NOT
- truth tables
- combining Boolean operators using AND, OR and NOT to two levels
- applying logical operators in appropriate truth tables to solve problems
- applying computing-related mathematics:
 - ✓ +
 - ✓ −
 - ✓ /
 - ✓ *
 - ✓ Exponentiation (^)
 - ✓ MOD
 - ✓ DIV

- Explain how data is represented by computer systems.
- Recognise and use appropriate mathematical operators.

- Group Presentations
- Individual Presentations
- Seneca Online activities
- Case Studies
- End of topic quiz
- End of term test
- Microsoft Teams collaborative activities.
- Programming Practical
- Mock Exams
- Home work
- Class Discussions
- Electronic and hardcopy Portfolio
- Topic Worksheets
- Past Paper question sheets

Spring 2	Translators and Facilities of Languages	• characteristics and purpose of different levels of programming language, including low level languages • the purpose of translators • the characteristics of an assembler, a compiler and an interpreter • common tools and facilities available in an integrated development environment (IDE): ✓ editors ✓ error diagnostics ✓ run-time environment ✓ translators.	• Describe the difference between low and high level languages. • Explain the advantages of using high level languages. • Explain how program instructions are encoded in low level languages • Explain why high level languages need to be translated • Explain the characteristics and use of ✓ an assembler ✓ a compiler ✓ an interpreter.	• Group Presentations. • Seneca Online activities • End of topic quiz • End of term test • Microsoft Teams collaborative activities. • Mock Exams • Home work • Class Discussions • Topic Worksheets • Past Paper question sheets
Summer 1	Revision and Examination preparations			• Group Presentations • One-on-one weak area(s) content review • Seneca Online activities • Past Paper question sheets