Year 11 Overview 2025-26 – *GCSE Computer Science*

<table>
<tr><th>Date</th><th>Wk</th><th>Week</th><th>Units Studied & Learning Outcomes</th><th>Key Concepts & Assessment</th></tr>
<tr><td colspan="5">8 weeks (20 Lessons) (38Days)</td></tr>
<tr><td>Tues 2-Sep
Y7 only
Wed-whole school</td><td>A</td><td>1</td><td rowspan="8">• Overview of Unit/No. lessons
Topic 6: Problem solving with programming

• Lesson Sequence of Content:
Lesson 1:2 - Subprograms
Lesson 3:4 - Local, global variables
Lesson 5:6 - Maths, time library methods
Lesson 7:8 - Problem solving

• Unit Learning Outcomes:
➢ understand and apply the fundamental principles and concepts of computer science, including abstraction, decomposition, logic, algorithms, and data representation
➢ analyse problems in computational terms through practical experience of solving such problems, including designing, writing and debugging programs.</td><td rowspan="8">• Foundational Concepts
Topic 6: Learning to program is a core component of a computer science course. Students should be competent at designing, reading, writing and debugging programs. They must be able to apply their skills to solve real problems and produce readable, robust programs.

• Key vocabulary
sequence, selection, repetition, iteration, procedure, function, parameter, return value, separation of concerns, local variable, global variable, math library, rounding, real number, time library, input, output, arithmetic operator

• Commentary
✓ Use sequence, selection (if, elif, else) and repetition (while) in programs
✓ Apply iteration to process items in one-dimensional structures
✓ Define and create procedures and functions with appropriate parameters and return values
✓ Understand and apply the difference between local and global variables
✓ Use math and time libraries to perform calculations and manage time
✓ Format and display numeric results using rounding functions and output formatting

• Foundational Concepts
✓ Recap of core content (e.g., software tools, design theory, programming logic)
✓ Transferable skills across projects
✓ Using retrieval strategies (flashcards, brain dumps, knowledge organisers)
✓ Metacognitive strategies (thinking about thinking)

• Assessment – x2 full exam papers
Papers will be generated to test the content covered to date, in Principles of Computer Science (written paper) & Application of Computational Thinking (coding paper)</td></tr>
<tr><td>8-Sep</td><td>B</td><td>2</td></tr>
<tr><td>15-Sep
(INSET Friday)</td><td>A</td><td>3</td></tr>
<tr><td>22-Sep</td><td>B</td><td>4</td></tr>
<tr><td>29-Sep</td><td>A</td><td>5</td></tr>
<tr><td>6-Oct</td><td>B</td><td>ST1</td></tr>
<tr><td>13-Oct</td><td>A</td><td>ST1</td></tr>
<tr><td>20-Oct</td><td>B</td><td>8</td></tr>
<tr><td>Half-Term</td><td colspan="4">7 weeks (17-18 lessons) (35 Days)</td></tr>
<tr><td>3-Nov</td><td>A</td><td>9</td><td rowspan="2">• Overview of Unit/No. lessons
Topic 3: Computers
Topic 4: Networks

• Lesson Sequence of Content:
Lesson 1:2 - Embedded systems
Lesson 3:4 - Low-level & high-level languages</td><td rowspan="2">• Foundational Concepts
Topic 3: Students must be familiar with the hardware and software components that make up a computer system.

Topic 4: Most computer applications in use today would not be possible without networks. Students should understand the key principles behind the organisation of computer networks.

• Key vocabulary</td></tr>
<tr><td>10-Nov</td><td>B</td><td>10</td></tr>
</table>

17-Nov	A	11	Lesson 5:6 - Translators Lesson 7:8 - The Internet of Things Lesson 9:10 - Packet switching Lesson 11:12 - TCP/IP 1 Lesson 13:14 - TCP/IP 2 Lesson 15:16 – End of unit assessment • Unit Learning Outcomes: ➢ understand the components that make up digital systems and how they communicate with one another and with other systems ➢ understand the impact of digital technology on wider society, including issues of privacy and cybersecurity	embedded system, general-purpose computer, hardware, software, low-level language, high-level language, instruction set, compiler, interpreter, translator, machine code, Internet of Things (IoT), security, privacy, power consumption, packet switching, IP address, router, TCP/IP stack, protocol, layer • Commentary ✓ Identify the characteristics and applications of embedded systems and compare them to general-purpose computers ✓ Understand and compare low-level and high-level languages, including their uses and instruction sets ✓ Explain the need for translators and compare how compilers and interpreters work ✓ Explore the IoT and discuss its security, privacy, and power-related issues ✓ Describe how data is transmitted via packet switching and the role of IP addresses and routers ✓ Understand how the TCP/IP stack enables communication across networks and explain the role of each protocol in the layers • **Assessment –** Informal assessment Sample assessment materials will be available to help prepare learners for the formal assessment
24-Nov	B	12		
1-Dec	A	13		
8-Dec	B	14		
15-Dec	A	15		
Christmas Holiday			6 weeks (15 lessons) (30 Days)	
5-Jan	B	16	• Overview of Unit/No. lessons Topic 5: Issues and impact • Lesson **Sequence** of Content: Lesson 1:4 - Environmental issues Lesson 5:6 - Intellectual property • Unit Learning Outcomes: ➢ understand the impact of digital technology on wider society, including issues of privacy and cybersecurity	• Foundational Concepts Topic 5: Students should be aware of the influence of digital technology and recognise some of the issues and the impact on wider society associated with its use. • Key vocabulary environmental impact, manufacture, energy consumption, e-waste, recycling, replacement cycle, digital technology, intellectual property (IP), copyright, patents, trademarks, open source, proprietary software • Commentary ✓ Explain how manufacturing and using digital devices affects the environment, and describe ways to reduce energy consumption ✓ Understand what e-waste is and discuss how recycling and short replacement cycles impact the environment ✓ Define intellectual property and explore how it is protected through copyright, patents, and trademarks, including the differences between open source and proprietary software
12-Jan	A	17		
19-Jan	B	ST2		
26-Jan	A	ST2	• Overview of Unit/No. lessons **Retrieval & Revision Practice** 7-8 lessons	• Foundational Concepts ✓ Recap of core content (e.g., software tools, design theory, programming logic) ✓ Transferable skills across projects ✓ Using retrieval strategies (flashcards, brain dumps, knowledge organisers) ✓ Metacognitive strategies (thinking about thinking) • **Assessment –** x2 full exam papers
2-Feb	B	20	• Unit Learning Outcomes: ➢ Identify and recall key concepts from the term's project work ➢ Apply knowledge to a range of unseen and exam-style tasks	

9-Feb	A	21	➢ Analyse strengths and gaps in understanding ➢ Improve performance through guided revision and peer/self-assessment	Papers will be generated to test the content covered to date, in Principles of Computer Science (written paper) & Application of Computational Thinking (coding paper)
Half-Term			6 weeks (15 lessons) (28 Days)	
23-Feb	B	22	• <u>Overview of Unit/No. lessons</u> Topic 1: Computational thinking Topic 6: Problem solving with programming Topic 2: Data • <u>Lesson **Sequence** of Content</u>: Lesson 1:2 - Trace tables Lesson 3:4 - Bubble sort Lesson 5:6 - Binary search Lesson 6:7 - Data structures Lesson 8:9 - Bitmaps Lesson 10:11 - Representation of sound Lesson 12:13 - Compression Lesson 14:15 – End of unit assessment • <u>Unit Learning Outcomes</u>: ➢ understand and apply the fundamental principles and concepts of computer science, including abstraction, decomposition, logic, algorithms, and data representation ➢ analyse problems in computational terms through practical experience of solving such problems, including designing, writing and debugging programs.	• Foundational Concepts Topic 1 & 6: Students are expected to develop a set of computational thinking skills that enable them to design, implement and analyse algorithms for solving problems. Students should be competent at designing, reading, writing and debugging programs. They must be able to apply their skills to solve real problems and produce readable, robust programs. Topic 2: Computers use binary to represent different types of data. Students are expected to learn how different types of data are represented in a computer. • Key vocabulary trace table, bubble sort, binary search, iteration, condition, one-dimensional list, two-dimensional list, index, traverse, record, field, bitmap, pixel, resolution, colour depth, image size, analogue, digital, amplitude, sample rate, bit depth, sample interval, ADC (analogue-to-digital converter), fidelity, compression, lossy, lossless, storage, bandwidth • Commentary ✓ Practise reading and using trace tables to predict the outcome of algorithms by tracking variable values ✓ Understand how bubble sort and binary search work, and how to recognise, trace, and modify code that implements them ✓ Explore the use and efficiency of one- and two-dimensional data structures, including reverse traversals and structured output ✓ Learn how bitmap images and sound are represented in binary, with key calculations for file size based on resolution, colour depth, and sample properties ✓ Explain compression methods, why they're used, and compare lossy vs lossless techniques with their advantages and limitations • **Assessment –** Informal assessment Sample assessment materials will be available to help prepare learners for the formal assessment
2-Mar	A	23		
9-Mar	B	24		
16-Mar	A	25		
23-Mar	B	26		
30-Mar (finish Wednesday 1st April)	A	27		
Easter Holiday			5 weeks (?? lessons) (24 Days)	
20-Apr	B	28	• <u>Overview of Unit/No. lessons</u> Topic 5: Issues and impact • <u>Lesson **Sequence** of Content</u>:	• Foundational Concepts Topic 5: Students should be aware of the influence of digital technology and recognise some of the issues and the impact on wider society associated with its use. • Key vocabulary

<table>
<tr><td>27-Apr</td><td>A</td><td>29</td><td rowspan="5">Lesson 1:2 - AI, machine learning & robotics
Lesson 3:4 - Personal data / Privacy
Lesson 5:6 - Data protection & computer misuse

• Unit Learning Outcomes:
➢ understand the impact of digital technology on wider society, including issues of privacy and cybersecurity</td><td rowspan="5">AI (Artificial Intelligence), machine learning, robotics, algorithmic bias, digital footprint, passive data, active data, personal data, privacy, data misuse, identity theft, Data Protection Act, Computer Misuse Act, data subject, algorithm, regulation, accountability, ethics

• Commentary
✓ Understand the definitions and real-world applications of AI, machine learning, and robotics, including where they are used and potential ethical/safety implications
✓ Explore the concept of algorithmic bias, how it emerges from training data, and consider how society might regulate emerging tech responsibly
✓ Identify examples of digital footprints, how personal data is gathered, and weigh up the risks and rewards of data sharing
✓ Analyse privacy concerns and understand complex issues around data ownership and identity protection
✓ Gain clarity on UK legislation like the Data Protection Act and Computer Misuse Act, and apply this to real-world examples of data misuse and protection

• Assessment – Informal assessment
Sample assessment materials will be available to help prepare learners for the formal assessment</td></tr>
<tr><td>4-May (Bank holiday Mon)</td><td>B</td><td>GCSE</td></tr>
<tr><td>11-May</td><td>A</td><td>GCSE</td></tr>
<tr><td>18-May</td><td>B</td><td>GCSE</td></tr>
<tr><td>Half-Term</td><td colspan="3">7 weeks (?? lessons) (35 Days)</td></tr>
<tr><td>1-Jun</td><td>A</td><td>GCSE</td><td rowspan="6">• Overview of Unit/No. lessons
Retrieval & Revision Practice remaining lessons

• Unit Learning Outcomes:
➢ Identify and recall key concepts from the term’s project work
➢ Apply knowledge to a range of unseen and exam-style tasks
➢ Analyse strengths and gaps in understanding
➢ Improve performance through guided revision and peer/self-assessment</td><td rowspan="6">• Foundational Concepts
✓ Recap of core content (e.g., software tools, design theory, programming logic)
✓ Transferable skills across projects
✓ Using retrieval strategies (flashcards, brain dumps, knowledge organisers)
✓ Metacognitive strategies (thinking about thinking)

• Assessment – x2 full exam papers
Papers will be generated to test the content covered to date, in Principles of Computer Science (written paper) & Application of Computational Thinking (coding paper)</td></tr>
<tr><td>9-Jun</td><td>B</td><td>GCSE</td></tr>
<tr><td>16-Jun</td><td>A</td><td>GCSE</td></tr>
<tr><td>23-Jun</td><td>B</td><td>GCSE</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td colspan="5">(Total: 190 Days)</td></tr>
</table>