Year 8 Overview 2026-27 – *Computer Science & TechMedia*

Date	Wk	Week	Units Studied & Learning Outcomes	Key Concepts & Assessment
8 weeks (11-12 Lessons) (37 Days)				
Wed 2-Sep Tues Y7 only Thursday-whole school	A	1	• Overview of Unit/No. lessons **2D Animation** 6-7 lessons • Lesson **Sequence** of Content: Lesson 1 – Idea generation Lesson 2 – Planning the animation Lesson 3 – Preparing assets Lesson 4 – Creating an animation Lesson 5 – Robot animation Lesson 6 – End of unit assessment • Unit Learning Outcomes: ➢ To know what is meant by a frame-by-frame animation. ➢ To produce sector-specific planning materials. ➢ To know what is meant by using assets in a media product. ➢ To know and apply your understanding of different animation techniques. ➢ To test/check if the animation meets the client requirements. ➢ To export the animation into a suitable file format.	• Foundational Concepts In this unit, learners will learn how to create a 2D animation by applying a range of animation techniques such as frame by frame animation, tweening and the effective use of frame rate. Learners will create and export a multi-layered animation into a format that can be played as a standalone file or as a moving image within a web page. • Key vocabulary Animation, Frame-by-frame, Tweening, Timeline, Keyframe, Frame rate, Storyboard, Assets, Concept art, Moodboard, Layers, Export, File format, Client brief, Playback • Links to the Key Stage 4 curriculum ✓ BTEC Creative Media ✓ BTEC Digital Information Technology • Commentary ✓ Introduction to animation (the concept of using frames) ✓ Complete a mind map and moodboard to generate ideas for the final animation ✓ Create a storyboard and concept art for the animation and its elements. ✓ Create assets for your animation. ✓ Practise creating animation in specialist software. • **Assessment -** Formal feedback will be given ✓ Complete end of unit assessment ✓ Mop-up of any missing work once the assessment is complete.
7-Sep	B	2		
14-Sep (INSET Friday)	A	3		
21-Sep	B	4		
28-Sep	A	5		
5-Oct	B	6		
12-Oct	A	7		
19-Oct	B	8	• Overview of Unit/No. lessons **Gamemaker** 7-8 lessons	• Foundational Concepts This unit introduces students to game development using a visual programming environment. Students explore core programming principles such as event-driven development, object-based design, and conditional logic.
Half-Term			7 weeks (10-11 lessons) (35 Days)	
2-Nov	A	9	• Lesson **Sequence** of Content: Lesson 1 – Player movement Lesson 2 – Maze design Lesson 3 – Enemies Lesson 4 – Levels Lesson 5 – Scores and lives Lesson 6 – Ending the game Lesson 7 – End of unit assessment • Unit Learning Outcomes: ➢ Learn how to use a development tool for making 2D games.	The unit supports understanding of how user experience, difficulty, and gameplay mechanics influence the design and coding of interactive systems. Students learn how to break down game features into smaller components (decomposition) and apply logic to solve problems within a digital context. • Key vocabulary Sprite, Object, Event, Collision detection, Score, Lives, Level, Game loop, Conditional statement, Game mechanics, Debugging, Randomisation, User interface, Asset, Iteration • Links to the Key Stage 4 curriculum ✓ BTEC Creative Media
9-Nov	B	ST1 (core only)		
16-Nov	A	ST1 (core only)		

23-Nov	B	12	➢ Understand how to create a game environment and re-design it to make it more difficult. ➢ Understand the purpose of enemies and collectible objects within the context of a video game. ➢ Understand the purpose of adding levels to a video game. ➢ Understand to purpose of scores and lives in a video game. ➢ Understand the conditions required to lose or win in your game.	✓ Edexcel GCSE Computer Science • Commentary ✓ To understand the requirements of the game you're creating. ✓ To create a game environment for each level of your game and include collision detection. ✓ Create sprites and assets for your game. ✓ Add new challenges to increase difficulty and engagement in your game. ✓ To write code that will create random events within your game, calculate the score and lives as you play the game, and that ends the game when all levels are complete. • **Assessment -** Formal feedback will be given ✓ Complete end of unit assessment ✓ Mop-up of any missing work once the assessment is complete.
30-Nov	A	13		
7-Dec	B	14		
14-Dec	A	15		
Christmas Holiday			6 weeks (8-9 lessons) (30 Days)	
4-Jan	B	16	• Overview of Unit/No. lessons **Computer Systems** 6-7 lessons • Lesson **Sequence** of Content: Lesson 1 – Boolean logic Lesson 2 – Binary Lesson 3 – Computer hardware Lesson 4 – Software Lesson 5 – Networks Lesson 6 – End of unit assessment • Unit Learning Outcomes: ➢ Understand the concept of logic gates and the role they play in a computer system. ➢ Understand what is meant by the term 'binary'. ➢ Understand computer hardware that is used as part of a computer system. ➢ Understand the purpose of software in a computer system. ➢ Understand computer networks including the internet, and how they can provide multiple services, such as the world wide web.	• Foundational Concepts This unit introduces the core components of computer systems, focusing on how data is processed and stored. Students learn the basics of binary representation and logic gates, providing foundational knowledge for digital systems. The unit also explores hardware, software, and network structures, helping students understand the architecture of modern computing. These ideas support progression into computer science theory at KS4 and underpin key topics such as data representation, system architecture, and communication. • Key vocabulary Binary, Denary, Logic gate, AND, OR, NOT, Truth table, CPU, RAM, Input device, Output device, Operating system, Application software, Network, Internet, Router, Switch, Data packet, Bandwidth • Links to the Key Stage 4 curriculum ✓ BTEC Digital Information Technology ✓ Edexcel GCSE Computer Science • Commentary ✓ To identify different logic gates and complete its truth table. ✓ To convert between binary numbers and denary numbers. Perform binary arithmetic. ✓ Identify different parts of a computer (including hardware and software) and understand their purpose. ✓ To know what is meant by a network. To know how data is transmitted over a network. • **Assessment -** Formal feedback will be given ✓ Complete end of unit assessment ✓ Mop-up of any missing work once the assessment is complete.
11-Jan	A	17		
18-Jan	B	18		
25-Jan	A	19		
1-Feb	B	20		
8-Feb	A	21		

<table>
<tr><td colspan="2">Half-Term</td><td colspan="3">5 weeks (7-8 lessons) (24Days)</td></tr>
<tr><td>22-Feb</td><td>B</td><td>22</td><td rowspan="5">• Overview of Unit/No. lessons
Retrieval & Revision Practice 3-4 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="5">• 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 – ST2
✓ Retrieval activities (quizzes, hinge questions, mini whiteboard tasks)
✓ Peer/self-assessment using success criteria
✓ Teacher feedback on misconceptions
✓ A formal end-of-unit exam</td></tr>
<tr><td>1-Mar</td><td>A</td><td>23</td></tr>
<tr><td>8-Mar</td><td>B</td><td>24</td></tr>
<tr><td>15-Mar</td><td>A</td><td>ST2</td></tr>
<tr><td>22-Mar
finish on Thursday 25th</td><td>B</td><td>ST2</td></tr>
<tr><td colspan="2">Easter Holiday</td><td colspan="3">7 weeks (10-11 lessons) (34 Days)</td></tr>
<tr><td>12-Apr</td><td>A</td><td>27</td><td rowspan="7">• Overview of Unit/No. lessons
Python programming 6-7 lessons

• Lesson Sequence of Content:
Lesson 1 – Output data
Lesson 2 – Input data
Lesson 3 – Selection
Lesson 4 – Iteration
Lesson 5 – Practical project
Lesson 6 – End of unit assessment

• Unit Learning Outcomes:
➢ Write programs that accept and respond appropriately to user input
➢ Write programs that make appropriate use of variables and constants
➢ Write programs that make appropriate use of primitive data types
➢ Write programs that make appropriate use of sequencing, selection, iteration
➢ Know the difference between a syntax error and logic error and know how to identify them in a program</td><td rowspan="7">• Foundational Concepts
In this unit, students will develop foundational skills in text-based programming using Python. They will understand the use of inputs, outputs, variables, data types, and basic programming constructs such as sequence, selection, and iteration. Students will also be introduced to debugging by identifying and correcting syntax and logic errors. These concepts underpin core programming knowledge and form the basis for further study at GCSE level.

• Key vocabulary
Input, Output, Variable, Constant, Data type, String, Integer, Casting, Sequence, Selection, Iteration, IF statement, FOR loop, WHILE loop, Syntax error, Logic error, Debugging, Condition, Expression

• Links to the Key Stage 4 curriculum
✓ Edexcel GCSE Computer Science

• Commentary
✓ Output data using print function; Input data using the input function.
✓ Use of constants and variables to store data
✓ Use of casting to change the data type.
✓ To know how to write an IF statement using a two-step & three-step procedure.
✓ To understand the difference between a counter-controlled loop (FOR) and condition-controlled loop (WHILE); write a basic FOR and WHILE loop.

• Assessment - Formal feedback will be given
✓ Complete end of unit assessment
✓ Mop-up of any missing work once the assessment is complete.</td></tr>
<tr><td>19-Apr</td><td>B</td><td>28</td></tr>
<tr><td>26-Apr</td><td>A</td><td>29</td></tr>
<tr><td>3-May
(Bank holiday Mon)</td><td>B</td><td>30</td></tr>
<tr><td>10-May</td><td>A</td><td>31</td></tr>
<tr><td>17-May</td><td>B</td><td>32</td></tr>
<tr><td>24-May</td><td>A</td><td>33</td></tr>
</table>

<table>
<tr><th>Half-Term</th><td colspan="5">6 weeks (8-9 lessons) (29 Days)</td></tr>
<tr><td>7-Jun</td><td>B</td><td>34</td><td rowspan="6">• Overview of Unit/No. lessons
Spreadsheets 6-7 lessons

• Lesson Sequence of Content:
Lesson 1 – Working with cells
Lesson 2 – Formatting
Lesson 3 – Formulas and functions
Lesson 4 – Charts
Lesson 5 – Practical project
Lesson 6 – End of unit assessment

• Unit Learning Outcomes:
➢ To understand the use of cells in spreadsheets
➢ To understand the purpose of formatting a spreadsheet
➢ To understand the purpose of formulas and functions.
➢ To understand the purpose of charts and why they are used to represent data.
➢ To demonstrate your practical knowledge of Spreadsheets.</td><td rowspan="6">• Foundational Concepts
Students will understand how spreadsheets can be used to store, process, and present data. They will explore how to apply formatting to enhance readability, use formulas and functions to perform calculations, and create charts to communicate information visually. These skills form a foundation for data literacy and are directly applicable to key Stage 4 coursework in vocational IT subjects.

• Key vocabulary
Cell, Worksheet, Cell reference, Formula, Function, SUM, AVERAGE, MIN, MAX, Merge cells, Formatting, Chart, Graph, Bar chart, Pie chart, Line graph, Data label, Axis, Legend, Spreadsheet model

• Links to the Key Stage 4 curriculum
✓ BTEC Digital Information Technology

• Commentary
✓ Understand why spreadsheets are used to store data.
✓ Using formatting techniques such as merge cells, text alignment and number formats.
✓ Using formulae to calculate sales and profits.
✓ Using functions to show highest, lowest and average values.
✓ Create a range of charts to visually represent data for a given scenario.
✓ To appropriately label each chart to ensure its fit for purpose.

• Assessment - Formal feedback will be given
✓ Complete end of unit assessment
✓ Mop-up of any missing work once the assessment is complete.</td></tr>
<tr><td>14-Jun</td><td>A</td><td>35</td></tr>
<tr><td>21-Jun</td><td>B</td><td>36</td></tr>
<tr><td>28-Jun
(SJBF Inset)</td><td>A</td><td>37</td></tr>
<tr><td>5-Jul</td><td>B</td><td>38</td></tr>
<tr><td>12-Jul</td><td>A</td><td>39</td></tr>
<tr><td colspan="6">(Total: 190 Days)</td></tr>
</table>

Year 8 CCM Curriculum Review: Summary of Implementation and Impact Overview

This year, we delivered a well-sequenced Year 8 curriculum that deepened students' understanding of digital creativity, computing logic, and practical problem-solving. Our focus was to strengthen technical competence, introduce development tools, and further embed responsible digital practices, supporting progression toward Level 2 vocational and academic pathways.

What We Taught (Key Units)

- **2D Animation:** Creating frame-by-frame and tweened animations.
- **Game Design:** Developing interactive games using sprites, collisions, and scoring systems.
- **Computer Systems:** Understanding how binary, logic gates, hardware, software, and networks work.
- **Python Programming:** Writing and debugging text-based code with variables, selection, and loops.
- **Spreadsheets:** Using formulas, functions, and charts to model and analyse data.

What Students Learned

- How to design, test, and improve digital products using appropriate tools.
- Key computing knowledge, including how software and systems function.
- How to code using a textual programming language (Python).
- How to present and manipulate data using spreadsheets.
- Practical project planning, testing, and evaluation techniques.

Links to Other Subjects & Real Life

- Cross-curricular links with maths (logic and data), art (design), music (soundtrack), and online safety education.
- Use of industry-relevant tools like Python, animation software, GameMaker, and spreadsheet software.
- Projects that mirror real-world media and IT tasks, building foundational skills that reflect future digital careers and Level 2 qualification requirements.