Now that the revised curriculum has been taught, please consider the Implementation and Impact of the curriculum you taught. What changes might need to be made to the Curriculum Intent (See Curriculum Map and Overviews) in light of this year's experiences?

<table>
<tr><th colspan="5">Year 9 Overview 2026-27 – CS & TechMedia</th></tr>
<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 (8 Lessons) (37 Days)</td></tr>
<tr><td>Wed 2-Sep
Tues Y7 only Thursday-whole school</td><td>A</td><td>1</td><td rowspan="8">Overview of Unit/No. lessons
App Development 6 lessons

Lesson Sequence of Content:
• Lesson 1 - Creating assets
• Lesson 2 - Building the app
• Lesson 3 - Event-driven programming
• Lesson 4 - Completing your app
• Lesson 5 – Login system
• Lesson 6 - End of unit assessment

Unit Learning Outcomes:

• Preparing and/or creating assets.
• Adding assets to an app.
• To know what is meant by a 'landing page'.
• To be introduced to event-driven programming.
• To complete an app that successfully meets the client brief.</td><td rowspan="8">• Foundational Concepts
This unit is designed to introduce students to the exciting world of app development and digital graphics. Students will learn the fundamentals of creating mobile applications and designing visually appealing digital graphics to enhance their apps. By the end of this unit, students will have the knowledge and skills to develop a simple mobile app and incorporate custom graphics into it.

• Key vocabulary
App, Assets, Repurpose, Landing page, Event-driven, condition, links, navigation, duplicate, credentials, PNG, File formats, Transparency.

• Links to the Key Stage 4 curriculum
BTEC Digital Information Technology
GCSE Computer Science

• Commentary
To create and/or repurpose your own assets to be used in the app.
To create a landing page and main menu screen for the GreenBite Express app.
To complete all screens and link them together.
To allow the user to input a username and password login into the app.

• Assessment – Formal feedback will be given
Complete weekly retrieval quizzes which recaps each lesson, feedback given related to performance score
Complete end of unit assessment
Mop-up of any missing work once the assessment is complete.</td></tr>
<tr><td>7-Sep</td><td>B</td><td>2</td></tr>
<tr><td>14-Sep
(INSET Friday)</td><td>A</td><td>3</td></tr>
<tr><td>21-Sep</td><td>B</td><td>4</td></tr>
<tr><td>28-Sep</td><td>A</td><td>5</td></tr>
<tr><td>5-Oct</td><td>B</td><td>6</td></tr>
<tr><td>12-Oct</td><td>A</td><td>7</td></tr>
<tr><td>19-Oct</td><td>B</td><td>8</td></tr>
<tr><td>Half-Term</td><td colspan="3">7 weeks (7 lessons) (35 Days)</td></tr>
<tr><td>2-Nov</td><td>A</td><td>9</td><td rowspan="2">Overview of Unit/No. lessons
Python Programming extended 7 lessons

Lesson Sequence of Content:
• Lesson 1 – Data types, structures and casting
• Lesson 2 - Operators
• Lesson 3 – Variables, constants, inputs and outputs
• Lesson 4 – Basic programming constructs
• Lesson 5 – Turtle module
• Lesson 6 – Introduction to subprograms
• Lesson 7 - End of unit assessment</td><td rowspan="2">• Foundational Concepts
This unit is designed to introduce students to the core principles of Python programming, building their confidence in writing and understanding code. Pupils will explore key programming concepts, gradually developing the skills needed to create functional programs. Throughout the unit, students will complete a workbook containing a variety of tasks. Pupils will also be given the opportunity to attempt practical programming challenges to deepen their understanding and problem-solving abilities.

• Key vocabulary
Functions, Variables, Integer, String, Boolean, Logic, Arithmetic, Constant, Iteration, Selection, Procedure

• Links to the Key Stage 4 curriculum
GCSE Computer Science

• Commentary
To understand and use different data types, structures and casting methods in Python.</td></tr>
<tr><td>9-Nov</td><td>B</td><td>10</td></tr>
</table>

16-Nov	A	11	<u>Unit Learning Outcomes</u>: • Know the different variable types. • Know the different operators and when to use them. • Understand the difference between a constant and variable and when to use them. • Know the 3 programming constructs. • Understand how to use the Python turtle module. • Understand the programs can be structured using procedures and functions. <u>Overview of Unit/No. lessons</u> **Retrieval & Revision Practice** 3 lessons <u>Unit Learning Outcomes:</u> • Identify and recall key concepts from the KS3 computing curriculum • Apply knowledge to a range of unseen and exam-style tasks	To apply, arithmetic, comparison and logical operators within programs. To use constructs such as sequence, selection and iteration effectively. To write and call simple subprograms to make code more modular and efficient. To create graphical outputs using the Turtle module for creative coding tasks. • **Assessment** – Formal feedback will be given Complete weekly retrieval quizzes which recaps each lesson, feedback given related to performance score Complete end of unit assessment Mop-up of any missing work once the assessment is complete. • 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)
23-Nov	B	12		
30-Nov	A	13		
7-Dec	B	14		
14-Dec	A	15		
Christmas Holiday			6 weeks (6 lessons) (30 Days)	
4-Jan	B	16	• Analyse strengths and gaps in understanding • Improve performance through guided revision and peer/self-assessment	• **Assessment -** ✓ Retrieval activities (quizzes, hinge questions, mini whiteboard tasks) ✓ Peer/self-assessment using success criteria ✓ Teacher feedback on misconceptions ✓ A formal end-of-unit exam
11-Jan	A	ST1		
18-Jan	B	ST1		
25-Jan	A	19	<u>Overview of Unit/No. lessons</u> **Spreadsheets extended** 6 lessons <u>Lesson **Sequence** of Content:</u> • Lesson 1 – Formlae and formatting • Lesson 2 – Validation • Lesson 3 – Conditional formatting	• Foundational Concepts This unit introduces pupils to the use of spreadsheets to solve real-world problems. They will learn how to create and use formulas, functions, and data tools to organise and analyse information effectively. Pupils will explore how spreadsheets can model systems such as budgets or performance trackers, and they will apply logical thinking to compare different ways of sorting and managing data.

Now that the revised curriculum has been taught, please consider the Implementation and Impact of the curriculum you taught. What changes might need to be made to the Curriculum Intent (See Curriculum Map and Overviews) in light of this year's experiences?

<table>
<tr><td>1-Feb</td><td>B</td><td>20</td><td rowspan="2">• Lesson 4 – Charts
• Lesson 5 – Functions
• Lesson 6 - End of unit assessment

<u>Unit Learning Outcomes:</u>
• To understand the use of cells in spreadsheets.
• To understand the purpose of formatting a spreadsheet.</td><td rowspan="2">Throughout the unit, pupils will create and refine spreadsheet-based projects with a focus on usability, accuracy, and clear design for a given audience.

• Key vocabulary
Cell, Cell reference, Row, Column, Formatting, Alignment, Wrap text, Formula, Function, Chart, Legend, Axis.

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

• Commentary
To understand why spreadsheets are used to store data.
To understand different ways a spreadsheet can be formatted to look professional.</td></tr>
<tr><td>8-Feb</td><td>A</td><td>21</td></tr>
<tr><td>Half-Term</td><td colspan="3">5 weeks (5 lessons) (24Days)</td></tr>
<tr><td>22-Feb</td><td>B</td><td>22</td><td rowspan="4">• 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.

<u>Overview of Unit/No. lessons</u>
Python Programming – Text Adventure game 7 lessons

<u>Lesson Sequence of Content</u>:
• Lesson 1 Identify concepts of text adventure games
• Lesson 2 Create script for text adventure game
• Lesson 3 Create flowchart for text adventure game
• Lesson 4 & 5 Code text adventure game</td><td rowspan="4">To understand the difference between formulas and functions and how to use them correctly.
To create a range of charts to visually represent data, appropriately labelling them.
To use a wide range of technical skills to update an existing spreadsheet.

• Assessment – Formal feedback will be given
Complete weekly retrieval quizzes which recaps each lesson, feedback given related to performance score
Complete end of unit assessment
Mop-up of any missing work once the assessment is complete.

• Foundational Concepts
This unit is designed to introduce students to the process of designing and creating a text adventure game using Python programming. Pupils will explore the key features and mechanics of text adventure games, developing their understanding of planning, scripting, and program flow. Throughout the unit, students will design their own game by creating a script and flowchart before applying their programming skills to code a functional text adventure game in Python. Pupils will also have opportunities to receive and provide peer feedback, helping them to refine their work and improve their problem-solving and computational thinking skills. The unit concludes with an assessment that allows students to demonstrate their understanding of game design and programming concepts.</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>25</td></tr>
</table>

22-Mar finish on Thursday 25th	B	26	• Lesson 6 Peer feedback • Lesson 7 End of unit assessment Unit Learning Outcomes: • Know the key features and conventions of a text adventure game. • Know how to use sequencing, selection and iteration within a program. • Understand how variables can be used to track player progress and choices.	• Key vocabulary Functions, Variables, Integer, String, Boolean, Logic, Arithmetic, Constant, Iteration, Selection, Procedure • Links to the Key Stage 4 curriculum ✓ GCSE Computer Science
Easter Holiday			7 weeks (7 lessons) (34 Days)	
12-Apr	A	27	• Understand how to use conditional statements to create branching pathways in an adventure game. • Know how to test and debug a text adventure game to identify and correct errors. • Understand the importance of peer feedback when improving a program. Overview of Unit/No. lessons **Retrieval & Revision Practice** 3 lessons Unit Learning Outcomes: • Identify and recall key concepts from the KS3 computing curriculum Apply knowledge to a range of unseen and exam-style tasks	• Commentary To identify and analyse the key features of text adventure games. To plan a text adventure game using scripts and flowcharts. To apply programming constructs such as sequence, selection and iteration within a game. To use variables, inputs and outputs to create interactive gameplay experiences. To write and test Python code that responds to user choices and actions. To evaluate and improve a text adventure game through testing and peer feedback. • **Assessment –** Formal feedback will be given Complete weekly retrieval quizzes which recaps each lesson, feedback given related to performance score Complete end of unit assessment Mop-up of any missing work once the assessment is complete. • 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)
19-Apr	B	28		
26-Apr	A	29		
3-May (Bank holiday Mon)	B	30		
10-May	A	31		
17-May	B	32		
24-May	A	33		

Now that the revised curriculum has been taught, please consider the Implementation and Impact of the curriculum you taught. What changes might need to be made to the Curriculum Intent (See Curriculum Map and Overviews) in light of this year's experiences?

<table>
<tr><td>Half-Term</td><td colspan="3">6 weeks (6 lessons) (29 Days)</td></tr>
<tr><td>7-Jun</td><td>B</td><td>34</td><td rowspan="6">• Analyse strengths and gaps in understanding
• Improve performance through guided revision and peer/self-assessment

• <u>Overview of Unit/No. lessons</u>
Vector Drawings 3 lessons

• <u>Lesson Sequence of Content:</u>
Lesson 1 – Basic Rotoscoping
Lesson 2 – Use of layers
Lesson 3 – End of unit assessment

• <u>Unit Learning Outcomes</u>:
➢ Apply rotoscoping techniques to trace and stylise visual elements accurately.
➢ Use layers effectively to organise and structure digital illustrations.
➢ Apply colour, gradients and shading techniques.
➢ Produce a low-poly style image using geometric shapes.</td><td rowspan="6">• Assessment -
✓ Retrieval activities (quizzes, hinge questions, mini whiteboard tasks)
✓ Peer/self-assessment using success criteria
✓ Teacher feedback on misconceptions
✓ A formal end-of-unit exam

• Foundational Concepts
This unit introduces pupils to digital vector artwork, starting with rotoscoping techniques to trace and stylise images. Pupils will use layers to organise their work and apply colour, shading, and gradients to enhance their designs. The unit ends with a low-poly drawing project, allowing pupils to create clean, geometric illustrations. Throughout, they will develop technical skills and reflect on how digital tools communicate visual ideas effectively.

• Key vocabulary
Vector, Rotoscoping, Anchor point, Layer, Opacity, Gradient, Shading, Stroke, Low-poly

• Links to the Key Stage 4 curriculum
BTEC DIT

• Commentary
✓ To use digital tools to trace and stylise images through rotoscoping techniques.
✓ To apply colour, gradients, and shading to enhance the look and depth of vector drawings.
✓ To create low-poly illustrations using geometric shapes and clean lines.
✓ To demonstrate accuracy and control when using a variety of vector drawing tools such as pen, shape and fill.

• Assessment - Formal feedback will be given
✓ Complete end of unit assessment
✓ Mop-up of any missing work.</td></tr>
<tr><td>14-Jun</td><td>A</td><td>ST2</td></tr>
<tr><td>21-Jun</td><td>B</td><td>ST2</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="4">(Total: 190 Days)</td></tr>
</table>

<u>Prompt Questions</u>

Year 9 CCM Curriculum Review: Summary of Implementation and Impact Overview

This year we delivered a Year 9 curriculum introducing students to a taster of the two options offered to students, Digital Information Technology and Computer Science. Our aim is to provide students with a broad and engaging introduction to the two subjects offered at Key Stage 4, enabling them to make an informed, confident and interest-led choice based on a clear understanding of the content, skills and expectations of each course.

What We Taught (Key Units)

- **App Development: Create a mobile app and incorporate custom graphics to it.**
- **Python Programming: Build confidence in writing and understanding code.**
- **Spreadsheets: Use formulas and functions to analyse information effectively.**
- **Text Adventure – Use of Python programming skills incorporated into pupil's text adventure games.**
- **Vector Drawings: Develop technical skills to produce digital vector artwork.**

What Students Learned

- **How to use digital tools confidently and creatively.**
- **Key computing ideas that prepare them for future study.**
- **How to code using a textual programming language (Python).**
- **Practical project planning, testing, and evaluation techniques.**
- **How to present and manipulate data using spreadsheets.**

Links to Other Subjects & Real Life

- **Connections to Art, Design & Technology and Maths.**
- **Use of real-world examples like mobile app development, graphic design, advertising, animation, and games design.**
- **Projects that mirror real-world media and IT tasks, building foundational skills that reflect future digital careers and Level 2 qualification requirements.**
- **Use of industry-relevant tools like Python, animation software and spreadsheet software.**