Subject: Computing Year: 8 Term: 1.1

Title of Scheme of Learning:	Overview of topic: *Programming (EduBlocks)* This unit introduces learners to text-based programming with Python. The lessons form a journey that starts with simple programs involving input and output, and gradually moves on through arithmetic operations, randomness, selection, and iteration. A range of pedagogical tools is employed throughout the unit, with the most prominent being pair programming, live coding, and worked examples. Key Skills to be covered: • Computational thinking problems • Programming constructs (Sequence, Selection, Iteration) • Text based programming • Programming using an IDE • Writing algorithms • Strings and Variables • Arithmetic and Number • Debugging SMSC/PSHE/CV/Catholic Social Teaching Overview: • Encourages creativity and imagination through coding, fostering a sense of enjoyment in creating programs (S) • Students learn about the ethical implications of programming, including the importance of writing code that is free from bias and respects user privacy (M) • Explore the global impact of technology and programming, understanding how different cultures use and contribute to technological advancements (CC) • Inspires students to think about how technology can be used to support and empower disadvantaged groups (CST) • Talk about the role that Britain has played in computing (CC)	How does this SOL relate to previous schemes? • Understand computational thinking concepts such as: decomposition, algorithmic design and abstraction (Y7) • Aware of difference between input and output (Y7) • Used a blocks based programming language such as scratch or MicroBit (Primary) (Y7) • Some experience of using variables to store data (Y7) Next topic: Introduction to programming using Python Links to Future Learning • Introduction to Algorithms (OCR Unit 2.1): EduBlocks introduces students to algorithmic thinking, a key concept in computational thinking that will be further developed when they study algorithms in OCR's curriculum. • Understanding Control Flow (OCR Unit 2.2): Through the use of sequence, selection, and iteration in EduBlocks, students gain practical experience with control structures, which are central to understanding the flow of a program in the OCR curriculum. • Developing Skills in Python (OCR Unit 2.2) As EduBlocks incorporates Python syntax, students build foundational skills in Python, directly preparing them for text-based Python programming tasks in later units.	Common Misconceptions in the unit: Print statements can only display numbers –Reality: Print statements can display strings, variables, and results of expressions, not just numbers. **The input function automatically converts user input to numbers:** The input() function always returns data as a string. Students need to explicitly convert it to the desired type (e.g., using int() for integers). **All arithmetic operations use the same symbol in programming as in math.**: Some symbols differ, such as * for multiplication and / for division, which can confuse students used to traditional mathematical notation. **The if statement can only check one condition:** if statements can check multiple conditions using elif and else clauses. **A loop must always start at 0:** Loops can start at any integer value, not just 0. The starting point can be specified by the programmer.

	PSHE (P): *Cultural Capital (CC)* *Spiritual, Moral, Social, Cultural (S/M/S/C)* *Catholic Social Teaching (CST)*		

Learning episodes (*inc nu of lessons*)	Learning Question(s) & Key Vocabulary	Content (*To include: What should be covered and planned time for students to carry out a response to feedback*)	Feedback & Assessment Points (*To include specific detail about what the assessments are & length of time for assessments etc*)

Input and Output (1)	How do user input and program output work together in programming?
	
	<u>Vocabulary</u>
	• Input
	• Output
	• Variable
	• Container
	• Print
	• Programming
	• String
		Students work through programming challenges on OneNote. They construct the code using the edublocks website. https://edublocks.org/
		
		Evidence to be recorded using screenshots which are uploaded to OneNote.
		
		To complete the programming challenges.
		
		Task 1 - Output Data
		Screenshot your solutions below:
		
		Challenge 1
		Create a program that will output the following nursery rhyme using multiple print statements.
		
		`Jack and Jill went up the hill`
		`To fetch a pail of water.`
		`Jack fell down and broke his crown,`
		`And Jill came tumbling after.`	

Task 2 - Input Data

Screenshot your solutions below:

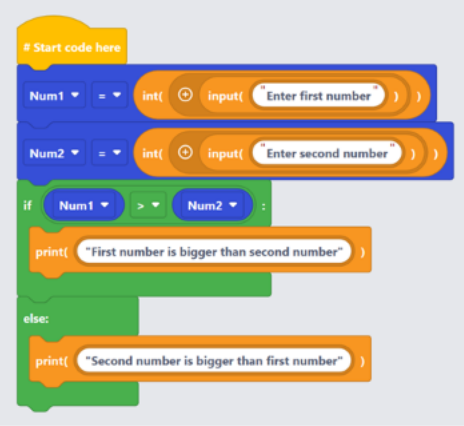
Challenge 1

Create a program that will allow the user to input a name which will then output the happy birthday song with their name.
 	
For example:
 	
`Happy birthday to you,`
`Happy birthday to you,`
`Happy birthday dear *name*`
`Happy birthday to you.`

Task 3 - Arithmetic Operations

Screenshot your solutions below

Challenge 1	**Challenge 2**
Create a program that will ask the user to enter two numbers and it calculates **Num1-Num2** and outputs the answer.	Create a program that will ask the user to enter two numbers and it calculates **Num1*Num2** and outputs the answer. **(* is used to multiply)**
Challenge 3	**Challenge 4**
Create a program that will ask the user to enter two numbers and it calculates **Num1/Num2** and outputs the answer. **(/ is used to divide)**	Create a program that asks the user to input three numbers and calculates their average. Output the average of the three numbers.
Challenge 5	
Create a program that prompts the user to enter the length and width of a rectangle. Calculate and output the area of the rectangle.	

Selection & Sequence (1)	How are the constructs of sequence and selection used in programming? <u>Vocabulary</u> • Selection • Sequence • Algorithm	Students work through programming challenges on OneNote. They construct the code using the edublocks website. https://edublocks.org/ Evidence to be recorded using screenshots which are uploaded to OneNote. To complete the programming challenges. Task 1 - IF - Else Below is an example of how to write an IF statement using IF-ELSE for only two outcomes.  Screenshot your solutions below: **Challenge 1** Create a program that will allow the user to input their age, it will then check whether they are **18 or over** or **under 18.** **Challenge 2** Create a program that will allow the user to input a test mark. If the score is **above 50** then its **pass** otherwise it's a **fail**.	

Task 2 - IF - ELIF - ELSE

Below is an example of how to write an IF statement using IF - ELIF - ELSE for more than two outcomes.

Screenshot your solutions below:

Challenge 1
Create a program that will allow the user to input the temperature of water. If it's **above 100** then its **boiling** and if its **below 0** then its **freezing.**

Challenge 2
Create a program that will allow user to input a score which will output one of the following grades: a. 30 is a Pass b. 60 is a Merit c. 90 is a Distinction Remember to include a condition for a fail!

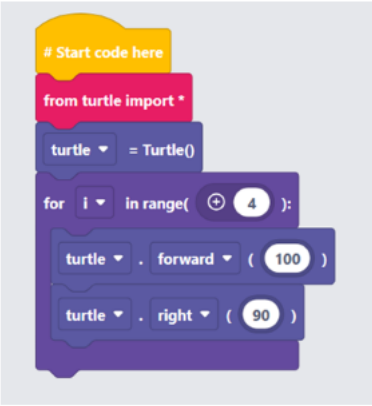
<table>
<tr>
<td></td>
<td></td>
<td>
<h2>Task 3 - Mastery Challenges</h2>
An example of how to create input on Edublocks


Screenshot your solutions below
<table>
<tr><th>Challenge 1 - Traffic Light Simulator</th><th>Challenge 2 - Number Category Checker</th></tr>
<tr><td>Create a program that simulates a traffic light. The program should ask the user for the current colour of the traffic light (red, yellow, or green) and provide the correct action to take (stop, slow down, or go).</td><td>Create a program that asks the user to enter a number. The program s then categorize the number as either "negative", "zero", or "positive". Additionally, if the number is positive, it should further categorize it as "small" (less than 10), "medium" (between 10 and 100), or "large" (gre than 100).</td></tr>
<tr><td></td><td></td></tr>
<tr><th>Challenge 3 - Simple Quiz</th><th>Challenge 4 - Shopping Discount Calculator</th></tr>
<tr><td>Create a simple quiz program that asks the user three multiple-choice questions. The program should check the user's answers and provide a score out of 3. It should also provide feedback based on the score (e.g., "Excellent", "Good", "Try again").</td><td>Create a program that calculates the total price of items in a shopping and applies a discount based on the total amount spent. The discount are as follows:
• No discount if the total is less than £50.
• 10% discount if the total is between £50 and £100.
• 20% discount if the total is over £100. The program should ask t to input the prices of three items, calculate the total, apply any discounts, and then display the final price.</td></tr>
</table>
</td>
<td></td>
</tr>
<tr>
<td>Iteration (1)</td>
<td>How is iteration used in programming?

<u>Vocabulary</u>
• Iteration
• Repetitio n
• For loop</td>
<td>Students work through programming challenges on OneNote. They construct the code using the edublocks website. https://edublocks.org/

Evidence to be recorded using screenshots which are uploaded to OneNote.

To complete the programming challenges.</td>
<td></td>
</tr>
</table>

	• Count-controlled	**Task 1 - Drawing Shapes** Below is an example of how to draw a shape.  Screenshot your solutions below: **Challenge 1** Create the following shapes: • Triangle (3-sided shape) • Pentagon (5-sided shape) • Hexagon (6-sided shape) • Octagon (8-sided shape) **Challenge 2** Create a new program that will draw a star shape. **Challenge 3** Create a new program that will print hello four times using a FOR loop. **Challenge 4** Create a program that will allow the user to input a number three times.	

<table>
<tr>
<td></td>
<td></td>
<td>
<h2>Task 3 - Mastery Challenges</h2>
Screenshot your solutions below
<table>
<tr><th>Challenge 1 - Even Number Printer</th><th>Challenge 2 - Multiplication Table Generator</th></tr>
<tr>
<td>Create a program that asks the user to input a number. The program should then use a for loop to print all even numbers from 1 up to that number.

Example:
If the user inputs `10`, the program should output:
Copy code
2
4
6
8
10</td>
<td>Create a program that asks the user to input a number and then uses a for loop to generate and display the multiplication table for that number up to 12.

Example:
If the user inputs `3`, the program should output:
python
3 x 1 = 3
3 x 2 = 6
...
3 x 12 = 36</td>
</tr>
<tr><th>Challenge 3 - Countdown Timer</th><th>Challenge 4 - Number Guessing Game</th></tr>
<tr>
<td>Create a program that asks the user to input a starting number for a countdown. The program should then use a while loop to count down to zero, printing each number, and finally print "Blast off!" when the countdown reaches zero.</td>
<td>Create a number guessing game. The program should generate a random number between 1 and 10. The user has three attempts to guess the number correctly. After each guess, the program should tell the user if their guess is too high, too low, or correct. If the user fails to guess in three attempts, the program should reveal the correct number.</td>
</tr>
</table>
</td>
<td></td>
</tr>
<tr>
<td>Practical Project (Programming Project)
4</td>
<td>How can we successfully complete a programming project?

<u>Vocabulary:</u>
• If-Else
• Parson's problem
• Integer
• Variable
• Loop</td>
<td>Start by completing the knowledge recall sheet

Students complete a programming project based on a client brief they have been given

You have been recruited as a programmer to develop a program for EarlyLearn that will help young children learn about different shapes and how to spell them.

EarlyLearn would you like the program to do the following:
Allow the child to enter the name of a shape.
If the name of the shape is spelt correctly then it will draw the shape.
If the name of the shape is spelt incorrectly then it will say "Sorry, this is not a shape".
The program should initially include two shapes (Square and Triangle) but can be extended further to include more shapes.

Students complete the programming project based on the client brief. Students need to work out how this will work through completing a parson's problem activity – worksheet provided</td>
<td>Practical Project - The creation of a programming project– Rubric used to assess and provide feedback
See assessment criteria in content box</td>
</tr>
</table>

<table>
<tr>
<td></td>
<td></td>
<td>
<h2>EduBlocks: Assessment criteria</h2>
<table>
<tr><th>Strand</th><th>Level 1</th><th>Level 2</th><th>Level 3</th><th>Level 4</th></tr>
<tr><td>Technical skills</td><td>Demonstrated a limited range of technical skills using suitable software.</td><td>Demonstrated a range of technical skills using suitable software.</td><td>Demonstrated a wide range of technical skills using suitable software.</td><td>Demonstrated a full range of technical skills using suitable software.</td></tr>
<tr><td>Practical project</td><td>May have an incomplete version of the programming project.</td><td>Created a solution that demonstrates an attempt at using selection and/or iteration.</td><td>Created a working solution that includes the use of IF-ELSE to successfully draw a shape.</td><td>Created a solution that includes the use of IF-ELIF-ELSE to successfully draw a range of shapes.</td></tr>
</table>
<table>
<tr>
<td>Technical skills – This is linked to lessons 1-3 and evidence will be recorded in the online workbook.

• Input and output
• Selection
• Iteration</td>
<td>Guidance of assessing the technical skills

Level 4 can be awarded if the student has fully completed all the tasks in their e-portfolio for lessons 1-3. The more parts of the e-portfolio that are incomplete, will correspond with the other levels.

Guidance of assessing the shape maker program

To save time marking, assess it as an algorithm rather than spending considerable time considering whether ‘will it work?’ Check of the criteria below when assessing the final solution.
• Import the turtle module
• Appropriate use of variables
• User input
• Selection statement – use if IF,ELIF and ELSE.
• Consideration for invalid/erroneous responses (i.e. not a recognisable shape name)
• Comparison operators – check correct use and conditions are set correctly.
• Counter-controlled iteration – use of FOR loops to draw shapes.</td>
</tr>
</table>
</td>
<td></td>
</tr>
<tr>
<td>Assessment 5</td>
<td>How can you demonstrate your knowledge and understanding of the topic?</td>
<td>Complete knowledge recall activity in preparation for end of unit assessment

Students will be given 40 minutes to complete an end of unit assessment to evaluate students’ knowledge at the end of the programming unit.</td>
<td>Complete summative end of unit assessment (link in the assessment folder)

Theoretical knowledge – A multiple-choice assessment</td>
</tr>
</table>
