Microsoft

DreamSpace

Python Programming Challenge

Post Video Resource Episode 3

Subject	Relevant Learning Outcomes	Key Skills
Computer Science (Leaving Cert)	- Design, write, test, and debug programs using a high-level language (Python). - Implement solutions using variables, loops, conditionals, and functions. - Apply problem-solving techniques in programming tasks.	- Critical & Creative Thinking (Developing solutions to coding problems) - Managing Information & Thinking (Logical reasoning and debugging) - Being Numerate (Applying computational thinking skills)
Mathematics	- Use computational thinking to solve numerical and logical problems. - Understand and apply algorithmic thinking to problem-solving.	- Being Numerate (Applying mathematical reasoning in Python) - Working with Others (Collaborating on problem-solving techniques)
Digital Technology (Junior Cycle)	- Explore how algorithms and programming can be used to solve problems. - Develop programs that demonstrate logic and structured thinking.	- Communicating (Explaining code in written and verbal formats) - Being Creative (Developing original coding solutions)

Learning Intentions

By the end of this activity, students will:

- Apply Python programming concepts to solve real-world problems.
- Develop logical thinking and debugging skills.
- Understand how loops, conditionals, and functions work in Python.
- Gain confidence in writing and troubleshooting their own Python code.

Success Criteria

Students will demonstrate success by:

- Writing Python scripts that correctly implement given problem statements.
- Debugging errors effectively and refining their code.
- Using appropriate programming structures (e.g., loops, functions) efficiently.
- Explaining their thought process and reasoning behind their solutions.

Activity Breakdown

Step 1: Warm-up Challenge (10-15 minutes)

- Before jumping into coding, ask students to predict outputs of short Python snippets involving:
  - Variable assignments
  - If-statements
  - Loops
  - Functions
- Example:

```python
x = 5
if x > 2:
    print("Greater")
else:
    print("Smaller")
```

Question: What will this print?

- Purpose: Reinforce fundamental concepts and prepare for coding tasks.

Step 2: Coding Challenges (40-50 minutes)

Students complete a tiered set of coding problems, progressing from basic to advanced.

💡 *Beginner Challenges (Fundamentals – 10-15 minutes each)*

1. Sum of a List
   - Write a program that takes a list of numbers and returns their sum.
   - Example Input: [3, 5, 2, 8] → Output: 18
   - Concepts: Loops, variables
2. Even or Odd?
   - Ask the user to enter a number and determine whether it's even or odd.
   - Example Input: 9 → Output: "Odd"
   - Concepts: Conditionals, user input

💡 Intermediate Challenges (Applying Concepts – 15-20 minutes each)

3. Guess the Number Game

- Generate a random number between 1-20, and let the user guess it, providing hints.
- Example Interaction

```
I'm thinking of a number between 1 and 20...
Enter your guess: 10
Too high! Try again.
Enter your guess: 4
Correct!
```

Concepts: Loops, conditionals, random module

1. Basic Calculator
    - Build a Python program that performs basic arithmetic operations (add, subtract, multiply, divide).
    - Example Input: 5 * 3 → Output: 15
    - Concepts: User input, functions

💡 Advanced Challenge (Creative Thinking – 30+ minutes)

5. Text-Based Adventure Game

- Students create a simple interactive text-based game where users make choices to progress through a story.
- Example

```
You find yourself in a dark cave. There are two tunnels.
Do you go left (L) or right (R)?
L
You encounter a sleeping dragon. Run (R) or Hide (H)?
```

- Concepts: Conditionals, functions, user input

Step 3: Debugging & Code Review (20 minutes)

1. Students exchange code with a peer to test and debug.
2. Checklist for debugging:
3. Does the program run without errors?
4. Are variables named clearly?
5. Are loops and conditionals used efficiently?

Step 4: Class Showcase & Reflection (15 minutes)

- Volunteers present their solutions, explaining their thought process and debugging challenges.
- Reflection Questions:
    - What was the most challenging part?
    - How would you improve your code?
    - Where could these coding skills be applied in real life?

Extension Activity (Optional)

- Students research a real-world Python application (e.g., AI, web development, cybersecurity) and write a short report or presentation on its impact.