**Week 1: Introduction to Computational Thinking & Python**

**Lesson Objectives:**

* Define computational thinking and its core components.
* Explain the Python programming environment.
* Write and run a basic Python program using print().

**Key Concepts:**

* Computational Thinking: Decomposition, Pattern Recognition, Abstraction, Algorithms
* Python set up and first program
* Introduction to Python syntax
* Using print() for output

**Suggested Agenda:**

1. **Warm-up Activity (5–10 min):**
   * Ask students to list the steps to brush their teeth — discuss as an example of algorithmic thinking.
2. **Direct Instruction (15 min):**
   * Slide deck or whiteboard intro to computational thinking concepts.
   * Introduce Python, show basic syntax and setup.
3. **Demonstration (10 min):**
   * Show how to write and run print("Hello, world!") using the Python interpreter or IDE.
4. **Guided Practice (15–20 min):**
   * Students follow along to write and run small print() statements (name, hobby, favorite food).
5. **Independent Practice (15 min):**
   * Complete one or more assignments from the list.
6. **Exit Ticket (5 min):**
   * Ask students to write one thing they learned and one question they still have.

**Materials Needed:**

* Projector or whiteboard
* Student access to Python (IDLE, VS Code, or Replit)
* Printed or digital lesson outline/assignments

**Optional Enrichment:**

* Read short article by Jeannette Wing: "What is Computational Thinking?"
* Ask students to find a real-world task they could break into computational steps.

**Resources:**

* [CS Unplugged: Computational Thinking](https://www.csunplugged.org/en/computational-thinking/)
* [Python.org - Getting Started](https://www.python.org/about/gettingstarted/)