Weekly Lesson Plans

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

* **Objectives:**
  + Understand computational thinking concepts
  + Install Python and run a basic script
* **Activities:**
  + Class discussion on decomposition, pattern recognition, abstraction
  + Python installation walkthrough
  + Students write their first program using print()
* **Assessment:**
  + Exit ticket: write a short program that prints a personal introduction

**Week 2: Variables, Data Types, and Input/Output**

* **Objectives:**
  + Declare variables and assign values
  + Use input() and print()
* **Activities:**
  + Live coding: user bio form
  + Pair activity: input and store favorite food, color, etc.
* **Assessment:**
  + Mini quiz on data types and inputs

**Week 3: Operators and Expressions**

* **Objectives:**
  + Apply arithmetic and logical operators
  + Understand operator precedence
* **Activities:**
  + Tip calculator activity
  + Quick puzzles using logical expressions
* **Assessment:**
  + Practice worksheet with expressions to evaluate

**Week 4: Conditional Statements**

* **Objectives:**
  + Use if, elif, and else statements
  + Create decision-making programs
* **Activities:**
  + Build a number guessing game
  + Scenario cards: decide outcomes using conditionals
* **Assessment:**
  + Programming task: grade checker based on score input

**Week 5: Loops Part 1 (While Loops)**

* **Objectives:**
  + Understand the use of while loops
  + Control infinite loops using conditions
* **Activities:**
  + Password entry loop activity
  + Live bug-fix session: infinite loop issues
* **Assessment:**
  + Code a countdown timer using while

**Week 6: Loops Part 2 (For Loops)**

* **Objectives:**
  + Use for loops with ranges and sequences
  + Understand nested loops
* **Activities:**
  + Multiplication table project
  + Loop tracing exercises
* **Assessment:**
  + Code review: student-created loop task

**Week 7: Functions and Modular Programming**

* **Objectives:**
  + Define and call functions
  + Understand parameters, return values, and scope
* **Activities:**
  + Write a function-based calculator
  + Function matching game: code snippets and descriptions
* **Assessment:**
  + Quiz on function structure and scope

**Week 8: Lists and Basic Data Structures**

* **Objectives:**
  + Create and modify lists
  + Use list methods and slicing
* **Activities:**
  + Grocery list manager project
  + Explore methods with sample lists
* **Assessment:**
  + Mini-lab: perform 5 different operations on a list

**Week 9: Problem Solving with Lists**

* **Objectives:**
  + Iterate through lists to compute and filter data
* **Activities:**
  + Build a grade analyzer
  + Loop-based filtering activity
* **Assessment:**
  + Code a program to find average and min/max in a list

**Week 10: Strings and Text Processing**

* **Objectives:**
  + Manipulate and analyze strings
* **Activities:**
  + Create a text analyzer
  + String method scavenger hunt
* **Assessment:**
  + Short-answer questions and program to count words

**Week 11: Introduction to Dictionaries**

* **Objectives:**
  + Create and use dictionaries to model real-world data
* **Activities:**
  + Build a phone book app
  + Dictionary comprehension practice
* **Assessment:**
  + Create a small dictionary app and explain its function

**Week 12: Algorithms and Algorithmic Thinking**

* **Objectives:**
  + Develop and trace basic algorithms
  + Write pseudocode
* **Activities:**
  + Linear vs binary search demo
  + Design and compare algorithms in small groups
* **Assessment:**
  + Pseudocode and code for simple search task

**Week 13: Debugging and Testing**

* **Objectives:**
  + Identify and fix common bugs
  + Write and apply simple test cases
* **Activities:**
  + Debugging scavenger hunt (find the bugs in code)
  + Test-driven fixes
* **Assessment:**
  + Debug a prewritten program

**Week 14: Introduction to Recursion**

* **Objectives:**
  + Understand and write recursive functions
* **Activities:**
  + Write factorial and Fibonacci recursively
  + Trace recursive calls with stack diagrams
* **Assessment:**
  + Write a recursive countdown function

**Week 15: Final Project Work**

* **Objectives:**
  + Apply computational thinking to a real problem
* **Activities:**
  + Students plan, code, and test final projects
  + Peer reviews
* **Assessment:**
  + Project checklist and work log

**Week 16: Project Presentations and Wrap-Up**

* **Objectives:**
  + Present a final programming project
  + Reflect on learning
* **Activities:**
  + Student presentations
  + Course retrospective
* **Assessment:**
  + Presentation rubric and reflection form