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

**✅ Key Concepts**:

* What is computational thinking?
* Decomposition, pattern recognition, abstraction, algorithms
* Python setup and first program

**📚 Resource**:

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

**🧠 Practice Problems**:

1. Write a program that prints your name and a fun fact about yourself.
2. Print a rectangle of asterisks using print("\*") multiple times.
3. Print a motivational quote using print().

**🛠️ Challenge**:

* Write a program that prints a welcome banner using ASCII art.

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

**✅ Key Concepts**:

* Variables, types: int, float, str, bool
* input() and print()
* Basic I/O flow

**📚 Resource**:

* [W3Schools: Python Variables](https://www.w3schools.com/python/python_variables.asp)

**🧠 Practice Problems**:

1. Ask for a user’s name and greet them.
2. Input two numbers, calculate their sum.
3. Ask for age, then print: "You’ll be [age+1] next year!"

**🛠️ Challenge**:

* Create a “Mad Libs” generator that asks for nouns, verbs, etc., and inserts them into a story.

**🗓️ Week 3: Operators and Expressions**

**✅ Key Concepts**:

* Arithmetic: +, -, \*, /, //, %, \*\*
* Comparison: ==, !=, >, <, etc.
* Logical: and, or, not

**📚 Resource**:

* [Real Python: Python Operators](https://realpython.com/python-operators-expressions/)

**🧠 Practice Problems**:

1. Calculate the area of a rectangle (length × width).
2. Determine if a number is even or odd using %.
3. Write an expression that evaluates if a user is a teenager (13–19).

**🛠️ Challenge**:

* BMI Calculator: Ask height and weight, compute and display BMI.

**🗓️ Week 4: Conditional Statements**

**✅ Key Concepts**:

* if, elif, else
* Nesting conditions
* Boolean logic in decisions

**📚 Resource**:

* [Programiz: Python if...else](https://www.programiz.com/python-programming/if-elif-else)

**🧠 Practice Problems**:

1. Ask for age; if 18+, print “You can vote.”
2. Ask for a test score and print A/B/C/D/F grade.
3. Determine the largest of three numbers.

**🛠️ Challenge**:

* Build a mini quiz: ask 3 questions, score the user's answers.

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

**✅ Key Concepts**:

* Looping with while
* Loop control: break, continue
* Infinite loops and loop conditions

**📚 Resource**:

* [Python Loops – While](https://www.w3schools.com/python/python_while_loops.asp)

**🧠 Practice Problems**:

1. Keep asking the user for a password until it’s correct.
2. Count from 1 to 10 using a while loop.
3. Make a countdown from 10 to 0 with print().

**🛠️ Challenge**:

* Build a number guessing game that keeps looping until the correct number is guessed.

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

**✅ Key Concepts**:

* for loops with range()
* Iterating through strings and lists
* Nested loops

**📚 Resource**:

* [Python for Loops](https://www.w3schools.com/python/python_for_loops.asp)

**🧠 Practice Problems**:

1. Print the first 10 square numbers.
2. Print all characters of a string, one per line.
3. Sum all numbers in a list using a loop.

**🛠️ Challenge**:

* Create a multiplication table (1–10) using nested loops.

**🗓️ Week 7: Functions and Modular Programming**

**✅ Key Concepts**:

* Writing and calling functions
* Parameters and return values
* Scope and modular code

**📚 Resource**:

* [Python Functions - Programiz](https://www.programiz.com/python-programming/function)

**🧠 Practice Problems**:

1. Write a function greet(name) that prints a greeting.
2. Write a function that takes a number and returns its square.
3. Write a function that checks if a number is even.

**🛠️ Challenge**:

* Build a calculator using functions: one for each operation (add, subtract, multiply, divide).

Would you like similar resources and practice sets for **Weeks 8–16** next?

Great! Here are **student resources** and **practice problems** for **Weeks 8–16** of the *Computational Thinking with Python*course. Each week includes:

* ✅ Key concepts
* 📚 Suggested reading or video
* 🧠 Practice problems
* 🛠️ Challenge activity (optional/advanced)

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

**✅ Key Concepts**:

* Creating and modifying lists
* Indexing, slicing
* List methods: append, remove, insert, etc.

**📚 Resource**:

* [W3Schools: Python Lists](https://www.w3schools.com/python/python_lists.asp)

**🧠 Practice Problems**:

1. Create a list of your favorite fruits and print the first and last item.
2. Add a new item to the list and remove one.
3. Reverse a list using slicing.

**🛠️ Challenge**:

* Build a shopping cart simulator where users can add/remove items and view the list.

**🗓️ Week 9: Problem Solving with Lists**

**✅ Key Concepts**:

* Iterating through lists
* Searching and sorting (concepts only)
* Intro to list comprehensions

**📚 Resource**:

* [Programiz: Python List Comprehension](https://www.programiz.com/python-programming/list-comprehension)

**🧠 Practice Problems**:

1. Write a program that prints all even numbers in a list.
2. Find the maximum and minimum values in a list.
3. Count how many times a value appears in a list.

**🛠️ Challenge**:

* Build a program to input grades, then display average, max, and min grades.

**🗓️ Week 10: Strings and Text Processing**

**✅ Key Concepts**:

* String operations: split, join, strip, find, replace
* String formatting
* Simple text processing

**📚 Resource**:

* [Real Python: String Methods](https://realpython.com/python-strings/)

**🧠 Practice Problems**:

1. Ask for a full name and print initials.
2. Count how many times a specific letter appears in a string.
3. Replace all spaces in a sentence with hyphens.

**🛠️ Challenge**:

* Write a text analyzer that counts words, sentences, and most frequent word.

**🗓️ Week 11: Introduction to Dictionaries**

**✅ Key Concepts**:

* Key-value pairs
* Adding, updating, deleting entries
* Looping through dictionaries

**📚 Resource**:

* [W3Schools: Python Dictionaries](https://www.w3schools.com/python/python_dictionaries.asp)

**🧠 Practice Problems**:

1. Create a dictionary with three contacts and their phone numbers.
2. Add a new contact and update one number.
3. Loop through all keys and values to print them.

**🛠️ Challenge**:

* Build a simple contact manager using a dictionary.

**🗓️ Week 12: Algorithms and Algorithmic Thinking**

**✅ Key Concepts**:

* Designing step-by-step solutions
* Pseudocode writing
* Search algorithms (linear, binary - conceptual only)

**📚 Resource**:

* [Khan Academy: Intro to Algorithms](https://www.khanacademy.org/computing/computer-science/algorithms)

**🧠 Practice Problems**:

1. Write pseudocode for making a sandwich.
2. Implement linear search to find an item in a list.
3. Describe how binary search would work on a sorted list.

**🛠️ Challenge**:

* Compare linear vs. binary search by counting steps taken in each.

**🗓️ Week 13: Debugging and Testing**

**✅ Key Concepts**:

* Types of errors (syntax, runtime, logic)
* Debugging with print() and tools
* Writing test cases

**📚 Resource**:

* [Python Errors and Exceptions](https://docs.python.org/3/tutorial/errors.html)

**🧠 Practice Problems**:

1. Fix a program with a syntax error (e.g., missing colon).
2. Identify and fix a logic error in a math calculation.
3. Add test cases for a function that returns the square of a number.

**🛠️ Challenge**:

* Given buggy code, fix all errors and write tests for it.

**🗓️ Week 14: Introduction to Recursion**

**✅ Key Concepts**:

* Recursive function structure (base case, recursive step)
* Examples: factorial, Fibonacci
* When recursion is helpful

**📚 Resource**:

* [Programiz: Python Recursion](https://www.programiz.com/python-programming/recursion)

**🧠 Practice Problems**:

1. Write a recursive function to calculate factorial.
2. Write a recursive function to compute the nth Fibonacci number.
3. Explain what happens if you forget a base case in recursion.

**🛠️ Challenge**:

* Write a recursive countdown timer from n to 0.

**🗓️ Week 15: Final Project Work**

**✅ Key Concepts**:

* Applying computational thinking end-to-end
* Planning, testing, debugging
* Modular design and functions

**📚 Resource**:

* Revisit key concepts from Weeks 1–14

**🧠 Practice Preparation**:

1. Brainstorm ideas using decomposition and abstraction.
2. Write a plan and pseudocode for your final project.
3. Test individual parts (functions/modules) separately.

**🛠️ Challenge**:

* Create a "Personal Assistant Bot" or "Mini Game" using all learned concepts.

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

**✅ Key Concepts**:

* Presentation and communication of code
* Reviewing computational thinking process
* Reflection and feedback

**📚 Resource**:

* [How to Present a Coding Project](https://medium.com/codex/how-to-present-your-code-project-48a9ff21e3cc)

**🧠 Practice Activity**:

1. Prepare a 2–3 minute walkthrough of your final project.
2. Practice explaining your problem-solving process clearly.
3. Write a reflection on what you learned in the course.

**🛠️ Challenge**:

* Peer-review a classmate’s code and give constructive feedback.

Would you like all of this in a PDF/printable format or as a student handout document?