# ✅ Week 1 — Thinking Like a Computer Scientist

**Theme:** Computational Thinking, Algorithms, Flowcharts, and Pseudocode

---

## 🎯 Learning Objectives

By the end of Week 1, students will be able to:

- ✅ Break real-world problems into algorithmic steps (decomposition)
- ✅ Translate logical ideas into **flowcharts** and **pseudocode**
- ✅ Understand the cycle: **Think → Plan → Code → Test**
- ✅ Explain how computers interpret instructions
- ✅ Write their first simple algorithm and Python “Hello World” program

---

## 📘 Core Concepts Explained (Deep Theory + Plain English)

### 🧠 *What is Computational Thinking?*

**Academic definition:** A systematic approach to problem-solving used in computer science that involves decomposition, pattern recognition, abstraction, and algorithmic design.

**Beginner explanation:** Computational thinking teaches you to solve problems step-by-step like a computer, not randomly like humans often do.

**Real-world analogy:** Think like a chef using a recipe. Every step must be in the right order, clear, reproducible, and leave no room for guessing.

---

## 🔍 *Core Thinking Components*

| Concept | Meaning | Real-World Example |
|---|---|---|
| Decomposition | Break big problem into smaller tasks | Cooking: prepare ingredients, heat pan, cook |
| Pattern Recognition | Notice repeated ideas | Buttons always click, login screens similar |
| Abstraction | Focus on important parts only | Driving: ignore trees, follow road signs |
| Algorithm Design | Write exact steps to solve | Recipe, ATM steps, Google Maps route |

---

##  *Algorithm vs Code*

| Term | Meaning |
|---|---|
| Algorithm | Human-readable steps to solve a problem |
| Code | Machine-readable instructions implementing algorithm |

💡 **First learn to think. Code comes after thought.**

---

## 🔄 *Flowcharts & Pseudocode*

**Flowchart** = visual plan **Pseudocode** = English-like algorithm

Flowchart example (boiling egg):

```
Start → Fill pot → Add eggs → Boil water → Wait 7 minutes → Stop heat
→ Serve → End
```

Pseudocode version:

```
Start
Fill pot with water
Add eggs
Boil water
Wait 7 minutes
Turn off heat
Serve eggs
End
```

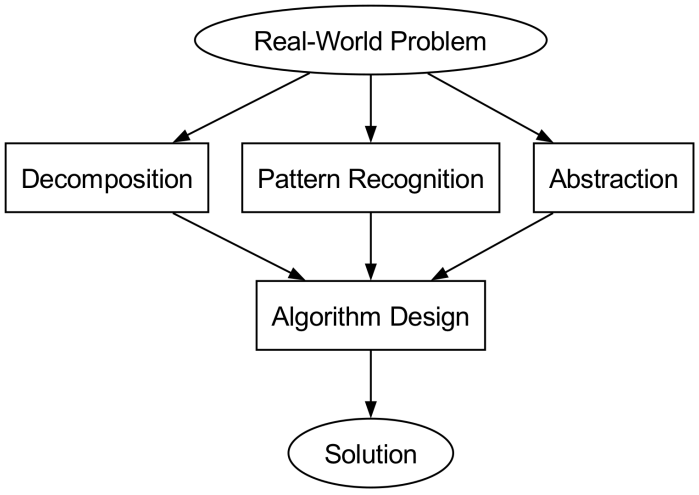
---

## 👨‍💻 *First Python Program*

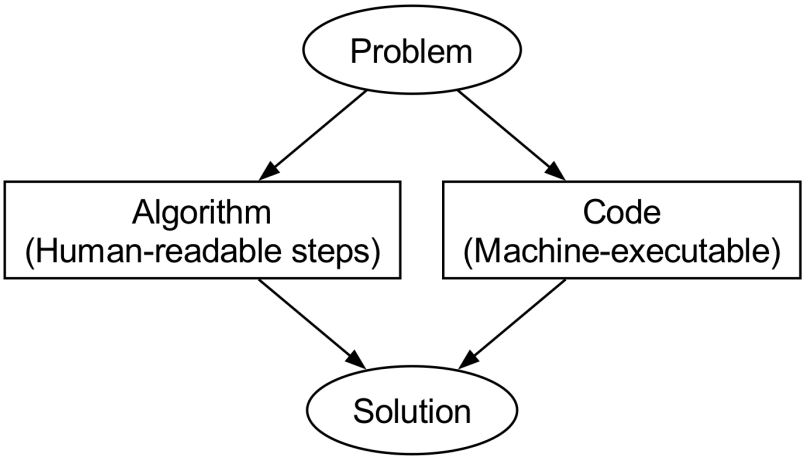
```python
# My first Python program
print("Hello, world!")
```

---

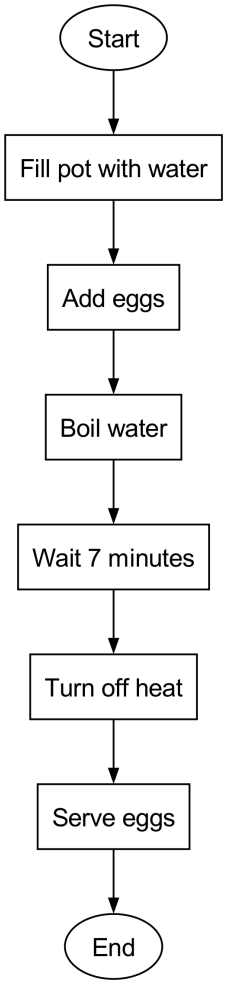
## Diagrams

- 

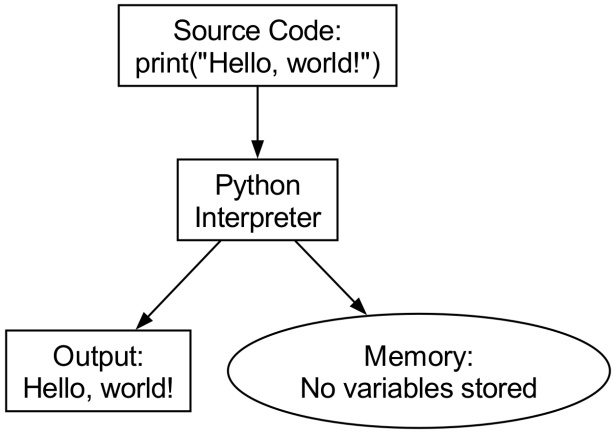

*Computational Thinking Components*

- 


*Algorithm vs Program Flow*




*Basic Flowchart for Simple Task*




*First Python Execution Diagram*

---

## 🏫 In-Class Exercises (3)

### ✅ *Exercise 1: Toothbrushing Algorithm*

Write a step-by-step algorithm to brush teeth.

### ✅ *Exercise 2: Flowchart for making tea*

Draw a flowchart for: **boil water → add tea → steep → add sugar → serve**

### ✅ *Exercise 3: First Python Output*

Write a program to print your name and reason for learning programming.

```
print("My name is ___")
print("I want to learn programming because ___")
```

---

## 🧠 Solutions to In-Class Exercises

### ✅ *Solution 1: Toothbrushing*

```
Pick toothbrush
Apply toothpaste
Wet toothbrush
Brush teeth for 2 minutes
Rinse mouth
Rinse brush
Store brush
```

### ✅ *Solution 2: Tea Flowchart (pseudocode)*

```
Start
Boil water
Add tea leaves
Wait 3-5 minutes
Add sugar
Serve
End
```

### ✅ *Solution 3: Python Output Example*

```python
print("My name is Sam")
print("I want to learn programming because I love building apps")
```

---

## 🏠 Take-Home Assignments (3)

### 1️⃣ *Algorithm — Preparing for school*

Write steps a student follows to get ready for school.

### 2️⃣ *Flowchart — Buying a bus ticket*

Include decisions (cash or card?).

### 3️⃣ *Python — Personal introduction script*

Print:

- Name
- Hometown
- Your goal for this course

---

## 📚 Weekly References

**Books**

- *How to Think Like a Computer Scientist — Python Edition*
- Jeannette Wing — *Computational Thinking (2006)*

**Online**

- MIT OpenCourseware — Intro to CS
- Python Official Docs — Getting Started

---

## ⚠️ Common Mistakes & Best Practices

| Mistake | Fix |
|---|---|
| Writing code without planning | Always sketch flowchart/ pseudocode first |
| Too many steps or too few steps | Steps must be clear, complete, logical |
| Using natural language, not algorithmic | Use simple short commands |
| Skipping debugging | Test small programs frequently |

---

## 🔮 Next Week Preview

Week 2 — Variables, Data Types & Memory

- What is data?
- Variables & assignment
- Memory model
- Input & output

---