Lesson 06: Pseudocode Puzzle - What Does This Code Do? (Low-Inquiry Version)

Lesson Title: Decoding Algorithms: Understanding What Code Does
Intended Grade Level(s): Grades 8-12 (adaptable)

I. Planning

Lesson Focus / Goals

The lesson aims to provide the following for students:

- Learn to read and interpret pseudocode (simplified programming language)
- Practice tracing code execution step-by-step using provided values
- Understand that code performs specific, predictable operations

Learning Objectives

By the end of the lesson, students will be able to:

- Read pseudocode and identify basic programming structures (loops, conditionals, variables)
- Trace code execution using a provided trace table
- State what the code does after the teacher explains it
- Explain why the code produces specific outputs for given inputs

Standards Alignment

Standards for Mathematical Practice (Common Core):

- **MP7** – Look for and make use of structure.
- **MP8** – Look for and express regularity in repeated reasoning.

NGSS Science and Engineering Practices:

- **Using Mathematics and Computational Thinking** – Students trace algorithmic processes step-by-step.
- **Developing and Using Models** – Students use trace tables to model computational processes.

Materials Needed

The following materials are used in the lesson:

- **Pseudocode handout** with three code samples (increasing complexity)
- **Trace table templates** for tracking variable values step-by-step
- **Teacher solution guide** showing complete traces and explanations

- **Projector** to display code and work through examples
- **Highlighters** to mark lines of code as they execute

II. Implementation

Lesson Flow

Before: (Launch – 8 min)

1. Ask: "Who here has written code or done any programming?"
2. Explain: "Even if you haven't programmed, you can understand what code does. Today we'll read 'pseudocode' - simplified code that's almost like English."
3. Show **simple example on projector:**

```
SET x = 5
SET y = 10
SET sum = x + y
DISPLAY sum
```

4. Ask: "What do you think this code does?"
5. Take 1-2 guesses, then explain: "It adds 5 and 10, stores the result in 'sum', and displays 15."
6. Introduce **trace table** concept: | Line | x | y | sum | Output | |——|—|—|——|———| | 1 | 5 | - | - | | | 2 | 5 | 10 | - | | | 3 | 5 | 10 | 15 | | | 4 | 5 | 10 | 15 | 15 |
7. Explain: "We'll trace each line to see exactly what happens."

During: (Explore – 14 min)

- Distribute **pseudocode handout** with three problems
- **Problem 1:** Simple loop

```
SET count = 0
REPEAT 5 times:
    ADD 1 to count
END REPEAT
DISPLAY count
```

- Students work individually to trace execution (3 min)
- Teacher shows solution and explains: "This code counts to 5. The loop repeats 5 times, adding 1 each time."
- Students fill in the complete trace table from teacher's model

- **Problem 2:** Conditional statement

```
SET number = 8
IF number > 10:
    DISPLAY "large"
ELSE:
    DISPLAY "small"
END IF
```

- Students trace (2 min)
- Teacher shows solution and explains: "This code checks if a number is greater than 10 and displays the appropriate message."
- Students copy the trace and explanation
- **Problem 3:** More complex loop (mystery operation)

```
SET n = 4
SET result = 1
REPEAT n times:
    MULTIPLY result by 2
END REPEAT
DISPLAY result
```

- Students trace (4 min)
- Teacher reveals: "This calculates 2 to the power of n. For n=4, it calculates 2^4 = 16."
- Teacher shows complete trace on board
- Explain each step: "Result starts at 1. Then multiply by 2 four times: 1→2→4→8→16"
- Students copy the full solution

After: (Discuss – 3 min)

- Ask: "What did all three code samples have in common?"
- Teacher summarizes: "They all had variables, followed steps in order, and produced predictable outputs."
- Emphasize: "Code isn't magic - it's just step-by-step instructions, like a recipe."
- Note: "Computer scientists and mathematicians write pseudocode before they write actual programs."

III. Assessment

Formative: During the lesson, monitor if students:

- Are tracing code line-by-line in the correct order
- Are updating variable values correctly in trace tables
- Can explain what each code sample does after teacher reveals the answer
- Successfully copy the teacher's complete solutions

Exit Ticket: Students trace one more simple code sample and state what it does:

```
SET x = 3
SET y = 7
SET answer = x * y
DISPLAY answer
```

(Expected: "This multiplies 3 and 7, displaying 21")

Peer/Self-Assessment: Students compare trace tables with a partner to verify they traced the steps correctly.

Student Handout

Name: _____ **Date:** _____

Pseudocode Puzzle: What Does This Code Do?

Instructions:
For each code sample, trace the execution line by line. Fill in the trace table. After your teacher explains what the code does, write the explanation.

Problem 1: Simple Loop

Code:

```
SET count = 0
REPEAT 5 times:
    ADD 1 to count
END REPEAT
DISPLAY count
```

Trace Table:

After Line	count	Output
1		
2 (1st time)		
2 (2nd time)		
2 (3rd time)		
2 (4th time)		

After Line	count	Output
2 (5th time)		
3		

What does this code do? (Fill in after teacher explains)

_____

_____

Problem 2: Conditional Statement

Code:

```
SET number = 8
IF number > 10:
    DISPLAY "large"
ELSE:
    DISPLAY "small"
END IF
```

Trace Table:

After Line	number	Output
1		
2 (condition)		
3 or 4		

What does this code do? (Fill in after teacher explains)

_____

_____

Problem 3: Mystery Operation

Code:

```
SET n = 4
SET result = 1
REPEAT n times:
    MULTIPLY result by 2
END REPEAT
DISPLAY result
```

Trace Table:

After Line	n	result	Output
1			
2			
3 (1st time)			
3 (2nd time)			
3 (3rd time)			
3 (4th time)			
4			

What does this code do? (Fill in after teacher explains)

Exit Ticket

Trace this code and explain what it does:

Code:

```
SET x = 3
SET y = 7
SET answer = x * y
DISPLAY answer
```

Trace Table:

After Line	x	y	answer	Output
1				
2				
3				
4				

What does this code do?