Multi-Agent AI Workflow Demo

This is a simulated multi-agent AI system designed to coordinate multiple AI agents to complete a user-facing task: writing an article. The system includes three agents:

1. Agent 1: Gathers relevant and recent data from simulated sources.
2. Agent 2: Verifies and fact-checks the gathered content.
3. Agent 3: Produces a well-structured and readable article based on verified content.

The orchestrator manages the workflow between agents, ensuring modularity and control.

```
import openai
from typing import Dict

# Simulated behavior instead of actual API calls (for demo purposes)
def agent_1_research(topic: str) -> str:
     return f"Collected recent and relevant information about '{topic}' from reliable sources."

def agent_2_verify(content: str) -> str:
      return f"Verified facts and corrected inaccuracies in the following content: {content}"

def agent_3_write(verified_content: str) -> str:
     return f"Final article draft: Based on {verified_content}. Written in a clear, user-friendly style."

# Coordinator class to orchestrate the agents
class MultiAgentOrchestrator:
    def __init__(self):
        self.logs: Dict[str, str] = {}

    def run_workflow(self, topic: str) -> str:
        self.logs['input_topic'] = topic

        # Step 1: Research
        research_output = agent_1_research(topic)
        self.logs['agent_1_output'] = research_output

        # Step 2: Verification
        verified_output = agent_2_verify(research_output)
        self.logs['agent_2_output'] = verified_output

        # Step 3: Writing
```

```
        final_article = agent_3_write(verified_output)
        self.logs['agent_3_output'] = final_article

        return final_article

    def get_logs(self):
        return self.logs

# Run example
if __name__ == "__main__":
    orchestrator = MultiAgentOrchestrator()
    result = orchestrator.run_workflow("The impact of AI in education")
    logs = orchestrator.get_logs()

    print("Workflow Output:\n")
    print(result)
    print("\n--- Logs ---")
    for step, output in logs.items():
        print(f"{step}: {output}")
```