🕵️ Problem Scenario: Secret Agent Intel Management System

You have been hired as a software developer for a **Top-Secret Intelligence Agency**. The agency needs a small system to securely store agent information in binary files and maintain mission notes in text files.

Task Requirements

1. **Agent Records (Binary File - `agents.dat`):**
   - Create an `Agent` class with attributes like `id` (agent code number) and `name` (cover name).
   - Allow the user to **add multiple agents** and store them in a **binary file**.
   - Implement a feature to **read and display all stored agents**, as if the agency is pulling up a classified dossier.
2. **Mission Notes (Text File - `missions.txt`):**
   - Create a `MissionFile` class for text file operations.
   - Allow agents to **write mission notes or intelligence reports** into a text file.
     - Writing continues until the agent types `~` (indicating the end of the message).
   - Implement a feature to **read and display all mission notes**, simulating a debriefing session.
3. **Menu-Driven Interface:**
   - When the program starts, display the following menu:

     ```
     ===== SECRET INTEL SYSTEM =====
     1. Add Agent Record
     2. View All Agents
     3. Write Mission Notes
     4. View Mission Notes
     5. Exit
     ===============================
     Enter your choice:
     ```

   - Perform the corresponding task using binary or text file handling.

🎯 Expected Learning Outcomes

By solving this, you will:

- Learn how to **store structured data securely** using binary files.
- Practice handling **free-form data** (mission notes) in text files.
- Gain experience designing a **menu-driven system**.

- Understand the difference between **binary vs. text file operations** in C++.
