Insurance Management System (OOP Based)

Below is a clean, professional, exam / lab–ready PROBLEM STATEMENT for the given C# program, followed by the Expected Output.
This is written exactly in the style teachers expect for practical exams, assignments, or viva questions.

Problem Statement

Title: Insurance Management System using Advanced OOP Concepts in C#

Objective

Design and implement a console-based Insurance Management System in C# that demonstrates the use of advanced object-oriented programming concepts including:

- Sealed classes
- Abstract base classes
- Inheritance and method overriding
- Method hiding using `new` keyword
- Init-only properties
- Encapsulation with validation
- Indexers
- Runtime polymorphism

Problem Description

You are required to build an Insurance System for an insurance company that manages different types of insurance policies. The system must follow real-world business rules and strict object-oriented design principles.

System Requirements

1. Security Module

- Create a sealed class that handles user authentication.
- This class must not be inherited.
- It should display a confirmation message after authentication.

---

2. Base Insurance Policy

- Create an abstract base class named `InsurancePolicy`.
- The class should contain:
  - An init-only property for Policy Number
  - A validated premium property (premium must be greater than zero)
  - A policy holder name
- Include:
  - A virtual method to calculate the premium
  - A method to display generic policy information

---

3. Derived Insurance Types

a) Life Insurance

- Inherit from `InsurancePolicy`

- Override the premium calculation method by adding a fixed life-insurance charge
- Demonstrate method hiding by redefining the policy display method

b) Health Insurance

- Inherit from `InsurancePolicy`
- Override the premium calculation method
- Seal the overridden method to prevent further modification

---

4. Policy Directory (Indexer Implementation)

- Create a class that stores multiple insurance policies.
- Use a List internally to store policies.
- Implement indexers to:
    - Access policies by index
    - Access policies by policy holder name

---

5. Main Program Execution

The main program should:

- Authenticate the user
- Create life and health insurance policies
- Store them in the policy directory
- Retrieve policy details using indexers
- Demonstrate:
    - Runtime polymorphism using overridden methods
    - Method hiding using base and derived class references

Expected Output

User authenticated successfully

Amit

102

Life Premium: 5500

Health Premium: 8000

Life Insurance Policy

Insurance Policy

Explanation of Output

1. Authentication message
    - Displayed by the sealed security class.
2. Indexer by index
    - Displays the holder name of the first policy (`Amit`).
3. Indexer by name
    - Displays the policy number of the policy holder named `Neha`.
4. Runtime polymorphism
    - Correct premium calculation based on actual object type.
5. Method hiding demonstration

Calling `ShowPolicy()` using a derived reference prints:
Life Insurance Policy

-

Calling the same method using a base-class reference prints:
Insurance Policy