1. The "Bank Vault" (Private vs. Public)

Goal: Understand how private protects data and public methods provide controlled access.

- **Task:** Create a class BankAccount.
- **Requirements:**
  - A private double balance.
  - A public method Deposit(double amount) that adds to the balance.
  - A public method GetBalance() that returns the current amount.
- **Challenge:** Try to change the balance directly from your Main method. (Spoiler: It should throw a compiler error!)

2. The "Library Book" (Properties & State)

Goal: Practice using different access levels for getters and setters.

- **Task:** Create a class Book.
- **Requirements:**
  - public strings for Title and Author.
  - A public string ISBN that can be read by anyone but **only set from inside the class** (use private set;).
- **Challenge:** Create a constructor that sets the ISBN when the book is first created.

3. The "Global Counter" (Static Members)

Goal: Learn how static variables belong to the class itself, not individual objects.

- **Task:** Create a class User.
- **Requirements:**
  - A static int userCount variable.
  - In the constructor of User, increment userCount by 1.
- **Challenge:** Create three different User objects in your Main method, then print User.userCount. It should show 3.

4. The "Math Wizard" (Static Classes & Methods)

Goal: Create a utility class that doesn't need to be "newed up."

- **Task:** Create a static class Calculator.
- **Requirements:**
  - A static method Add(double a, double b).
  - A static method CalculateCircleArea(double radius). Use Math.PI for the calculation:

$$Area = \pi r^2$$

- **Challenge:** Try to instantiate the class using new Calculator(). You’ll see that C# prevents this because the class is static.

5. The "ID Generator" (Combining Concepts)

Goal: Use a static field to provide unique, read-only data to instances.

- **Task:** Create a class Employee.
- **Requirements:**
  - A private static int nextId = 101;.
  - A public readonly int EmployeeId;.
  - In the constructor, assign EmployeeId = nextId and then increment nextId.
- **Challenge:** Create two employees and print their IDs. They should be 101 and 102 respectively.