Exercise 1: The Global Counter

Goal: Track how many objects are created, but initialize a "Starting ID" only once.

- Create a class Product.
- Add a static int field named NextId.
- Use a **static constructor** to initialize NextId to a random number between 1000 and 2000.
- In the **instance constructor**, assign the current NextId to an instance property ProductId, then increment NextId.
- **Test:** Create three products and print their IDs to verify they are sequential starting from the static random value.

Exercise 2: Application Settings Loader

Goal: Simulate loading configuration data that shouldn't change.

- Create a static class AppConfig.
- Add static properties: Theme (string), MaxLoginAttempts (int), and LastBootTime (DateTime).
- Use a **static constructor** to set Theme to "Dark", MaxLoginAttempts to 5, and LastBootTime to DateTime.Now.
- In your Main method, print these settings. Then, use Thread.Sleep(1000) and try to print the LastBootTime again to prove it remains the same as when the app started.

Exercise 3: The Math Utility Constant

Goal: Perform a complex calculation once to be used by all instances.

- Create a class CircleCalculator.
- Define a static double field named PrecomputedValue.
- In the **static constructor**, calculate a complex value (e.g., $Math.PI * Math.E$) and print "Static Initialization Done".
- In the **instance constructor**, print "Instance Created".
- **Constraint:** Create 5 instances. Verify that "Static Initialization Done" only appears once, while "Instance Created" appears five times.

Exercise 4: Validating Static Data

Goal: Ensure global data is valid before the program uses a class.

- Create a class SystemValidator.
- Define a static string named EnvironmentName.
- In the **static constructor**, check if the current day is a weekend (DateTime.Today.DayOfWeek).
    - If it is a weekend, set EnvironmentName to "Maintenance Mode".
    - Otherwise, set it to "Production".
- Create an instance method ShowStatus() that prints the EnvironmentName.

Exercise 5: Static vs Instance Order

Goal: Observe the execution flow (excellent for debugging logic).

- Create a class Logger.
- Add a static string log trace.
- In the **static constructor**, append " [Static] " to the trace.
- In the **instance constructor**, append " [Instance] " to the trace.
- In your Main method:
    1. Print the trace *before* creating an object.
    2. Create one object.
    3. Print the trace again.
- **Requirement:** Explain why the trace looks different in step 1 versus step 3.