Exercise 1: The Basic Profile

Goal: Initialize a basic anonymous type and access its properties.

- Create an anonymous type named car with the properties Make, Model, and Year.
- Assign it values (e.g., "Tesla", "Model 3", 2024).
- Print a single string to the console that says: "I drive a 2024 Tesla Model 3." using string interpolation.

Exercise 2: Property Name Inference

Goal: Understand how the compiler borrows names from existing variables.

- Declare two local variables: string city = "Tokyo"; and int population = 14000000;.
- Create an anonymous type called location that uses these variables **without** explicitly defining new property names (e.g., new { city, population }).
- Print the property names and values to verify that the compiler inferred the names city and population.

Exercise 3: Read-Only Constraint

Goal: Prove the immutability of anonymous types.

- Create an anonymous type representing a bankAccount with AccountNumber and Balance.
- Try to update the Balance to a new value on the next line.
- **Task:** Note the specific compiler error message you receive. Why does this happen?

Exercise 4: Non-LINQ Collections

Goal: Use anonymous types within an array.

- Create an implicitly typed array (using var) that contains three different anonymous objects representing products.
- Each object should have a Name (string) and Price (decimal).
- Use a standard foreach loop to iterate through the array and calculate the **total price** of all products.

Exercise 5: The "With" Expression (C# 10+)

Goal: Learn how to "mutate" an anonymous type by creating a copy.

- Create an anonymous type originalPoint with X = 10 and Y = 20.
- Use the with keyword to create a new anonymous type called movedPoint that copies originalPoint but changes Y to 50.
- Print both points to show that the original remains unchanged.