Exercises: The ref Keyword

Focus: Modifying an existing state.

1. **The Swapper:** Write a method Swap(ref int a, ref int b) that exchanges the values of two integers.
2. **The Bank Teller:** Create a method Withdraw(ref decimal balance, decimal amount). If the balance is sufficient, deduct the amount; otherwise, print an error.
3. **String Sanitizer:** Write a method CleanText(ref string input) that removes all whitespace and converts the string to lowercase.
4. **The Wrapper:** Create a method WrapValue(ref int value, int min, int max). If the value exceeds max, set it to min. If it falls below min, set it to max.
5. **Array Element Updater:** Write a method DoubleElement(ref int element) and call it while iterating through an array to double every number in that array.

Exercises: The out Keyword

Focus: Returning data and status codes.

1. **The Divider:** Write a method Divide(int dividend, int divisor, out int quotient, out int remainder).
2. **Coordinate Finder:** Create a method GetPoint(out int x, out int y) that prompts a user for input and "returns" the coordinates via out parameters.
3. **Area and Perimeter:** Write a method for a rectangle that takes length and width as inputs and provides both area and perimeter via out.
4. **Login Validator:** Create a method ValidateLogin(string input, out string errorMessage). If the input is empty, the error message should be "Username required."
5. **The Multi-Parser:** Write a method that takes a string and tries to parse it into an int, a double, and a bool simultaneously using out for each result.

Exercises: The in Keyword

Focus: Efficiency and read-only safety.

1. **The Logger:** Write a method Log(in string message) that prints a timestamp followed by the message. Ensure the method treats the message as read-only.
2. **Vector Magnitude:** Create a large struct called Vector3D (with X, Y, Z). Write a method CalculateMagnitude(in Vector3D vector) that returns the length without copying the struct.
3. **Tax Calculator:** Write a method ApplyTax(decimal price, in decimal taxRate). The method should return the total price but must not be able to modify the taxRate.
4. **Config Reader:** Create a method DisplaySettings(in AppConfig config) where AppConfig is a struct containing multiple strings. The method should only read and print them.
5. **Distance Checker:** Write a method IsWithinRange(in Point target, in Point player, float range) to check if the player is close to the target without duplicating the Point data.