# Moq Hands-On Exercise

## Questions & Answers

1. **What is Mocking in Unit Testing?**
   * Creating simulated objects that mimic real objects
   * Used to isolate the system under test from its dependencies
   * Allows testing without relying on real implementations
2. **Why use Mocks?**
   * Test code in isolation
   * Avoid external dependencies (databases, APIs, etc.)
   * Make tests faster and more reliable
   * Simulate error conditions easily
3. **Key Moq Features Used:**
   * Mock<T>: Creates a mock object
   * Setup(): Configures mock behavior
   * Verify(): Checks if methods were called
   * It.IsAny<T>(): Matches any argument of type T
4. **Dependency Injection Benefits:**
   * Enables loose coupling
   * Makes code more testable
   * Allows easy swapping of implementations
   * Supports mocking in unit tests
5. **Test Structure:**
   * Arrange: Create and configure mocks
   * Act: Execute the method under test
   * Assert: Verify results and interactions