# Moq

## Exercise Title:

Mocking with Moq in C#

## Objective:

To create unit-testable code using dependency injection and mock external services using Moq.

## Project Structure:

- CustomerCommLib: Contains the IMailSender interface and its concrete implementation MailSender.  
- CustomerComm: The class under test.  
- CustomerCommTests: NUnit Test project using Moq to isolate and test CustomerComm.

## Key Concepts:

- Dependency Injection (DI): Used to pass IMailSender to CustomerComm.  
- Mocking: Replaces real implementation with a mock object.  
- Testable Design: The method SendMailToCustomer() doesn’t rely on external SMTP server.

## Technologies Used:

- .NET Framework (net48)  
- NUnit  
- Moq  
- Microsoft.NET.Test.Sdk

## Testing Code Summary:

- Used [TestFixture], [SetUp], and [Test] attributes.  
- Mocked SendMail() to always return true.  
- Asserted that SendMailToCustomer() returns true.

## Output:

CustomerCommTests test succeeded  
Test summary: total: 1, failed: 0, succeeded: 1, skipped: 0

Refer to the terminal screenshot attached in your submission.