PARENT CODE

```
namespace MyApp
{
    public interface ILogger
    {
        void Log(string message);
    }

    public class Calculator
    {
        private readonly ILogger _logger;

        public Calculator(ILogger logger)
        {
            _logger = logger;
        }

        public int Add(int a, int b)
        {
            int result = a + b;
            _logger.Log($"Add called with {a} and {b}, result: {result}");
            return result;
        }

        public int Subtract(int a, int b)
        {
            int result = a - b;
            _logger.Log($"Subtract called with {a} and {b}, result: {result}");
            return result;
        }
    }
}
```

UNIT TEST CODE

```
using NUnit.Framework;
using Moq;
using MyApp;
namespace MyApp.Tests
{
    public class CalculatorTests
    {
        private Mock<ILogger> _mockLogger;
        private Calculator _calculator;
        public void Setup()
        {
            _mockLogger = new Mock<ILogger>();
            _calculator = new Calculator(_mockLogger.Object);
        }

        --TEST
        public void Add_ShouldReturnCorrectSum_AndLogMessage()
        {
            // Arrange
            int a = 5, b = 3;

            // Act
            int result = _calculator.Add(a, b);

            // Assert
            Assert.AreEqual(8, result);
            _mockLogger.Verify(logger => logger.Log(
                It.Is<string>(msg => msg.Contains("Add called with 5 and 3"))), Times.Once);
        }


        public void Subtract_ShouldReturnCorrectResult_AndLogMessage()
        {
            // Arrange
            int a = 10, b = 4;

            // Act
            int result = _calculator.Subtract(a, b);

            // Assert
            Assert.AreEqual(6, result);
            _mockLogger.Verify(logger => logger.Log(
                It.Is<string>(msg => msg.Contains("Subtract called with 10 and 4"))),
Times.Once);
        }
    }
}
```