1. Write Testable Code with Moq

Code:  
  
using Moq;

using NUnit.Framework;

// Interface to be mocked

public interface IUserRepository

{

    string GetUserById(int id);

}

// Class under test

public class UserService

{

    private readonly IUserRepository \_repository;

    public UserService(IUserRepository repository)

    {

        \_repository = repository;

    }

    public string GetUserName(int id)

    {

        var user = \_repository.GetUserById(id);

        return user ?? "Unknown";

    }

}

// Unit Test using Moq

[TestFixture]

public class UserServiceTests

{

    private Mock<IUserRepository> \_mockRepo;

    private UserService \_userService;

    [SetUp]

    public void Setup()

    {

        \_mockRepo = new Mock<IUserRepository>();

        \_userService = new UserService(\_mockRepo.Object);

    }

    [Test]

    public void GetUserName\_ReturnsUserName\_WhenUserExists()

    {

        // Arrange

        \_mockRepo.Setup(r => r.GetUserById(1)).Returns("Rahul");

        // Act

        var result = \_userService.GetUserName(1);

        // Assert

        Assert.AreEqual("Rahul", result);

    }

    [Test]

    public void GetUserName\_ReturnsUnknown\_WhenUserDoesNotExist()

    {

        // Arrange

        \_mockRepo.Setup(r => r.GetUserById(2)).Returns((string)null);

        // Act

        var result = \_userService.GetUserName(2);

        // Assert

        Assert.AreEqual("Unknown", result);

    }

}

output:  
