

## Student.cs

```
public class Student
{
    public int StudentId { get; set; }
    public string StudentName { get; set; }
    public string StudentEmail { get; set; }
    public long StudentPhone { get; set; }
}
```

## Subject.cs

```
public class Subject
{
    public int SubjectCode { get; set; }
    public string SubjectName { get; set; }
}
```

## Teacher.cs

```
public class Teacher
{
    public int TeacherId { get; set; }
    public string TeacherName { get; set; }
    public string TeacherDescription { get; set; }
}
```

## Test.cs

```
using NUnit.Framework;
using System.Collections.Generic;

[TestFixture]
public class Test
{
    [Test]
    public void StudentTest()
    {
        var studentList = new List<Student>();
        var student = new Student { StudentId = 1, StudentName = "Sam", StudentEmail = "sam@email.com", StudentPhone = 9876523456};

        studentList.Add(student);
        Assert.AreEqual(1, studentList.Count);
    }

    [Test]
    public void SubjectTest()
    {
        var subjectList = new List<Subject>();
        var subject = new Subject { SubjectCode = 101, SubjectName = "Biology"};
    }
}
```

```

        subjectList.Add(subject);
        Assert.AreEqual(1, subjectList.Count);
    }

    [Test]
    public void TeacherTest()
    {
        var teacherList = new List<Teacher>();
        var teacher = new Teacher { TeacherId = 1, TeacherName = "John",
TeacherDescription = "Biology Teacher"};

        teacherList.Add(teacher);
        Assert.AreEqual(1, teacherList.Count);
    }

```

## Repository.cs

```

public interface IStudentRepository
{
    Student GetById(int id);
}

public interface ISubjectRepository
{
    Subject GetById(int id);
}

public interface ITeacherRepository
{
    Teacher GetById(int id);
}

public class Repository
{
    private readonly IStudentRepository _studentRepository;
    private readonly ISubjectRepository _subjectRepository;
    private readonly ITeacherRepository _teacherRepository;

    public Repository(IStudentRepository studentRepository, ISubjectRepository
subjectRepository, ITeacherRepository teacherRepository)
    {
        _studentRepository = studentRepository;
        _subjectRepository = subjectRepository;
        _teacherRepository = teacherRepository;
    }

    public Student GetStudentById(int id)
    {
        return _studentRepository.GetById(id);
    }

    public Subject GetSubjectById(int id)
    {

```

```

        return _subjectRepository.GetByld(id);
    }

    public Teacher GetTeacherByld(int id)
    {
        return _teacherRepository.GetByld(id);
    }
}

```

## MockTest.cs

```

using NUnit.Framework;
using Moq;

```

```

[TestFixture]
public class MockTest
{
    [Test]
    public void GetByld()
    {
        // Arrange
        var studentRepositoryMock = new Mock<IStudentRepository>();
        studentRepositoryMock.Setup(repo => repo.GetByld(1)).Returns(new Student {
StudentId = 1, StudentName = "Alice", StudentEmail = "alice@email.com", StudentPhone =
9765425487 });

        var subjectRepositoryMock = new Mock<ISubjectRepository>();
        subjectRepositoryMock.Setup(repo => repo.GetByld(1)).Returns(new Subject {
SubjectCode = 1, SubjectName = "Math" });

        var teacherRepositoryMock = new Mock<ITeacherRepository>();
        teacherRepositoryMock.Setup(repo => repo.GetByld(1)).Returns(new Teacher {
TeacherId = 1, TeacherName = "John Doe", TeacherDescription = "Biology" });

        var repository = new Repository(studentRepositoryMock.Object,
subjectRepositoryMock.Object, teacherRepositoryMock.Object);

        // Act
        var student = repository.GetStudentByld(1);
        var subject = repository.GetSubjectByld(1);
        var teacher = repository.GetTeacherByld(1);

        // Assert
        Assert.NotNull(student);
        Assert.NotNull(subject);
        Assert.NotNull(teacher);

        Assert.AreEqual("Alice", student.StudentName);
        Assert.AreEqual("Math", subject.SubjectName);
        Assert.AreEqual("John Doe", teacher.TeacherName);
    }
}

```