# Design Principles Lab, John Lawless

Link to repository on github : https://github.com/dshirt/DesignPrincipleslabAssignment.git

2) In the “single” package, you have a class called “UserController”. It is violating “The Single Reasonability Principal”.

Q2a Explain how this is happening?

According to the lecture slides, “ A software module should have one and only one responsibility”.

All the code for User creation, user validation and storing the user is held in one class. This means the UserController has three responsibilities. It should be divided into three classes. One for creation, one for validation and another for storing the user. Each responsible for one thing.

2c. Do you notice any advantages to your code after you refactor?

The code is cleaner, easier to read. The class names now describe what each class does. It makes changing code much easier.

3) In the package “open”, examine the two classes “PhoneSubscriber and ISPSubscriber”

a. Refactor the two classes so that they follow the Open-Closed Principal

b. Do you notice any advantages to your code after you refactor?

Much easier to read. We reuse code instead of repeating it, i.e. we make use of polymorphism. If the company now wants to add other types of Subscriber they just need to extend the Subscriber class, that class shouldn’t be modified.