The University of the West Indies, St. Augustine
COMP 3607 Object Oriented Programming II
2018/2019 Semester 1
Lab Tutorial #4

This tutorial focuses on the SOLID design principles.

Learning Objectives:

- Reinforce your knowledge on each of the SOLID design principles
- Discuss examples of SOLID design principles in code snippets
- Refactor existing code to conform to one or more SOLID design principles

Section A: SOLID Knowledge

1. List the five SOLID design principles.
2. What are the key concepts of each of the SOLID design principles?
3. What are the consequences of violating each of the SOLID design principles?
4. Consider class A with the following operations:
   - Open a database connection
   - Fetch data from database
   - Write the data in an external file

(a) What is the issue with this class? _____

Suppose any of the following changes happens in future.

- New database
- Adopt ORM to manage queries on database
- Change in the output structure

(b) What happens to class A in each case?
(c) What are the implications (if any) of the first change?
(d) Which SOLID principle is being violated here?
(e) What is the solution to this problem according to the principle you've identified in (d) above?

Section B: Code Examples

5. Consider an Animal parent class:

```
public class Animal {
    public void makeNoise() {
        System.out.println("I am making noise");
    }
}
```

(a) Write code for two subclasses Dog and Cat so that we are able to replace an Animal object with the Dog or Cat without causing an error. A Dog and a Cat should make the appropriate noise for its behaviour.

(b) What happens when the following class is introduced?

```
class DeafDog extends Animal {
    @Override
    public void makeNoise() {
        throw new RuntimeException("I can't make noise");
    }
}
```

(c) Which principle is violated here?

(d) What is the solution to this problem according to the principle you've identified in (e) above?

6. In Android, there are multiple click listeners such as *OnClickListener* and *OnLongClickListener*. The *OnClickListener* specifies the method for a callback to be invoked when a view is clicked. The*OnLongClickListener* specifies the method for a callback to be invoked when a view has been clicked and held.

(a) What is the problem with having the following interface to specify the behaviour for these two listeners?

```
public interface MyOnClickListener {
    void onClick(View v);
    boolean onLongClick(View v);
}
```

(b) Which principle is violated here?

(c) Why does this principle suggest using two separate interfaces?

Section C: Simple Refactoring

7. Refactor the code examples in section B to adhere to the various principles

Section D: Additional Exercise

Suppose we have a system that handles authentication through external services such as Google, GitHub, etc. We would have a class for each service: *GoogleAuthenticationService*, *GitHubAuthenticationService*, etc. Now, let's say that some place in our system, we need to authenticate our user. To do that, as mentioned, we have several services available. Using code examples, explain two ways of handling this problem and discuss the strengths and weakness of each option citing the appropriate SOLID design principle(s) in play.