Course: CSC 340.05 Toe

Student: Steven McHenry, SFSU ID: 916931878

Teammate: None

Assignment Number: 01

Assignment Due Date & Time: 09-15-2019 at 11:55 PM

**Part A**

Prompt: Please choose 5 guidelines and discuss them in depth. For each guideline, use at least half a page for your discussion.

Answer:

Guideline 1 – Cohesion:

The basis of cohesion is how well the features and methods of a class belong together in a single class. Do these various members of the class work together when given the objective of accomplishing a task? The goal, in terms of Java classes, it to sustain a high level of cohesion throughout your project. When you are thinking about your program, it is a good first step to decide which classes will need to be made. These classes should have as narrow a set of objectives as possible. If you develop a general “actions” class for example that holds the various methods that you will need to use throughout your program, you are looking at an extremely low level of cohesion. The various methods within this class will likely have little to do with one another and thus will be better off in their own separate classes. Cohesion is good within a program because it makes your code easier to understand and more module, which can allow you to re-use these classes in other programs. A cohesive class is much easier to maintain and can also be scaled up when needed for the program.

Guideline 2 – Encapsulation

The overall goal of encapsulation is to hide the direct state of an object instance and only allow the interaction of that instance through methods which are specifically designed within that class. There are multiple benefits to this paradigm, but the biggest one of all is securing your data from unintended change. The best way to encapsulate your variables within a class is with the “private” modifier, which indicates that the variable cannot be accessed outside of the class. This requires the addition of “getter” and “setter” methods which, understandably, get the intended data of the object instance and set the object instance’s variables to a designated value. When referencing an instance’s fields you may need to use the “this” keyword to indicate that you are indeed intending to reference the current instance of that object. Another large benefit to encapsulation is the reduction of coupling it adds to your program. Any given class should not rely on the implementation of another class, and encapsulation ensures just that.

Guideline 3 – Consistency