**SOLID & GRASP write-up**

*Participants: Richard Kozyak, Eric Shao, Christian Chin, Khoa Bui, Alexander Wang, Brian Chen*

*In this doc file, we will identify three examples of GRASP principles being followed within our application.*

**Creator**

The creator pattern is used to guide the assignment of responsibilities to objects in object-oriented design, it simpler terms, to guide decisions about which class should be responsible for creating instances of another class. In this project, one of the examples for the use case of creator pattern is in WorkoutTracker.java. We **assign WorkoutTracker.java in ‘view’ the responsibility of creating instances of Workout.java in ‘model’.** WorkoutTracker.java follows the Creator pattern because it contains a collection of Workout objects and has the necessary information to initialize them. Below is the provided snippet of code:

A screenshot of a computer program

Description automatically generatedA screenshot of a computer program

Description automatically generated

**Single Responsibility Principle**

A screenshot of a computer program

Description automatically generatedEach class has a single responsibility. One of our example is in the **UserDatabase.java in ‘model’. It ensures that only one instance of these database classes is created throughout the application's lifecycle.** Doing so will prevent the issues that could arise from having multiple instances, such as data inconsistency, race conditions, and unnecessary memory usage. The getInstance() method is responsible for checking if the instance is null, and if so, it initializes the instance while synchronizing on the respective class to ensure thread safety and guarantees that only one instance is created.

**Information Expert**

The Information Expert Principle states that **responsibilities are assigned to the class that has the necessary information to fulfill it.** For instance, if a class holds data about a specific entity, it should also have methods to process that data. One of our examples is **User.java in ‘model’**. It processes information about user name, height weight, and gender, therefore, getUsername(), setUsername(), getUserId(), getName(), getHeight(), getWeight(), and getGender() are assigned to User.java in order to retrieve, as well as assign data to user instances.

A screenshot of a computer program

Description automatically generated