Week 9 – Group Activity – SOLID & GRASP:

Participants: Jesse Chen, Jeff Chow, Jayden Causey, Ignatius Djaynurdin, Rachel Kim, Henry Tang

For our week 9 group activity involving SOLID and GRASP principles, we created 3 different Java files that incorporate around 5 of those principles. The first is Project.java, which is a full java class with a constructor and 2 functions that add/remove tasks to a project object and 2 functions that add/remove team members to said project object. We also have a Task.java file, an interface used for other task Java classes that we did not implement for the assignment. The same applies to our TeamMember.java file which is an interface as well.

Within our program we use a couple of different principles. When it comes to GRASP principles, we mainly utilized Information Expert, Low Coupling, and High Cohesion. For Information Expert we assigned Project.java to take care of adding and removing tasks or team members because project objects have information storing who is within the team and what tasks need to be done. For Low Coupling, our java program has classes that are not very closely connected. In fact, since we only have 3 Java and none of them share too much data from variables to functions, we exhibit Low Coupling. Additionally, the few aspects of coupling we do use is mainly through the much less egregious Stamp Coupling, where we pass in Tasks or Team Member objects to fit our project's array lists storing them under their interfaces. Finally for the GRASP principles, we also apply High Cohesion because our project class is intended to be within its own area while the interfaces Team Member or Tasks will be used to extend into their own individual specific types of tasks or team members.

When it comes to SOLID principles, we also use a few. The first is "Open to Expansion, Closed to Modification." By using interfaces within our code, we open our code to have future Task or Team Member classes to extend our interfaces, thus allowing for expansion. At the same time, the interfaces themselves cannot be modified. The next is the idea of Single Responsibility. Essentially, our Task and Team Member interfaces exist with only one single responsibility, which is to store information regarding their respective names. In summary, our Java program applies the SOLID and GRASP principles of Information Expert, Low Coupling (and Stamp Coupling), High Cohesion, "Open to Expansion, Close to Modification" and Single Responsibility.