Assignment 3:

CS 3398

In-Class (and maybe some after class):

1. **Using Planning Poker to Prioritize/Rate Your Projects and**
2. **Using Three SOLID Principles to Partition Team Work**

(60 Team pts)

Due EOD, Tuesday, September 10

Scoring:

Planning Poker:

- **25 pts:** A project stories spread sheet with (at least) 2 stories per team member. Each project story has Planning Poker ratings for *Interest*, *Difficulty and Software Engineering Learning*. Points will be taken off or missing ratings or fewer than requested project stories.
- **5 pts**: A screen shot of Planning Poker with names of the team members and (some of the) ratings given to the stories.

Three SOLID Principles:

- **15 pts:** There is a least one file per team member with his/her code. 0 pts if any team members are missing (unless circumstances suggest that some team members couldn't participate).
- **10 pts:** The contents of each file must be explained using the 3 principles. If a principle did not apply, then just say so. This should be done by the team member responsible for the file. Other team members should agree with it.
- **5 pts:** A qualitative assessment of your solutions.
- To submit this homework, please send a single Slack notification per team in the general channel of the Team's Slack that announces that the team believes it is done with this assignment.

1. After obtaining from TRACS, unzip ThreeSolidPrinciples into a working directory. It is based upon the Workers and Robots example for Presentation 2 (also found on this website):[i]
   - https://www.oodesign.com/interface-segregation-principle.html
2. As a team, examine the single file with code: *ThreeSolidMain.java*
   - Notice that all interfaces and classes are defined in this single file.
3. Your team will add a Robot class as shown in Presentation 2 and then break up this code using the three SOLID principles of:
   - Open Close
   - Single Responsibility
   - Interface Segregation
   - And
     - Your desire to have everyone on the team work on the code.
4. This means if you have 4 team members, you will have at least 4 files. If you have 5 team members, you will have at least 5 files. Each team member is responsible for at least one file.
5. Please use the following guidelines:
   - The interfaces must be in (at least) one separate file.
   - You will need a separate main program file.
   - Please decide what you want to do with the Manager class. (EG: Where should it go?)

- When you add the Robot class, you will now have 3 related classes, Worker, SuperWorker and Robot. What should you do with these to make their implementation follow Open Close and Single Responsibility?

6. Explain your reasoning for your reorganizing the code. What principles were you using?
7. Please build your solution using Ant (see instructions in TRACS).
8. Have the team submit all the files together on the team Slack.

In our next meeting, we will try to place them into the Team GitHub while in class.

---

[i] Note that the website has two typos. In the "good example," the IWorker interface extends the Feedable and Workable interfaces. It should say that it extends the IFeedable and IWorkable interfaces.