Week 1 Notes

Intro to the field of Software Design (from Week 0)

1. What do we mean, when we say "Software Design is about people"?
Software design is aiming to make it clear for people to read instead of letting the software running faster.

2. What does "efficiency" refer to, when we talk about Object-Oriented Design?
How much time it cost to do something.

Intro to git (based on Week 1 lecture and first lab)

3. What is a Version Control System and why is it useful?
A system helping you track the changes made to the code. It is useful because with it you can always get the most updated code in time.

4. Briefly, what do each of the following git commands do?

`git clone`
Copy a remote repository to the computer (local).

`git pull`
Get the most updated version of the repository

`git add`
Add files to git.

`git commit -m "Replace this with a message"`
Record changes to local repository

`git push`
Upload the most updated version of the repository to the remote repository

`git checkout`
Switch to other branches to work on

`git branch`
Get or remove a branch and determine either the branch exists or not

`git merge`
Combine branches

5. Why is it important to make commit messages concise and informative?
Transfer clear message about changes made to group members.

6. Why do we use branches and pull requests?

Avoid mistakes uploaded to the main branch where others pull from and provide inconvenience when others work on the repository.

Intro to CRC (based on Week 1 lecture)

7. What is a CRC card? What is a CRC model?
A method to brainstorm the structure and relationship between classes and functions at the designing stage.

8. Briefly describe how a software development team can use CRC to design a program together?
Use the CRC cards to arrange the structure and relationship between classes and functions at the designing stage. Make it clear when assigning parts to members.

Intro to Java (based on Java readings and quizzes)

9. Why do we need a main method?
To run the file

10. What is a "constructor", "instance variable", "class variable", and "method"?
"constructor" is a method to create the instance of a class.
"instance variable" is a variable declared in a class
"class variable" is a static variable declared in a class
"method" is a storage of code which can be called.

11. Which package is the String class in? You can find this information on the Oracle website by searching "String java" on your favourite search engine.
java.lang

Week 0 Lab

12. List three things that stood out to you in the code reading exercise during the first lab. These can be about the code itself, differences between Java and Python, or even about how you felt the group discussion went.
1. Changes of grammar from python. 2. People are optimistically discussing in the breakout room. 3. Clear explanation from our TAs.

13. What parts of git or the GitHub website are you comfortable with now? What parts are you still unsure of? (We recommend you get in touch with us either through Piazza or office hours to address any problems you're facing!)
Pushing, pulling and commit. Auto grading is still causing my uncertainty.

Week 1 Lab

14. In the "abstract class or interface?" exercise, one of the examples was about Beverages (Tea, Juice, Coffee, etc.) and another had items that could be drunk (Tea, Coffee, Soup, etc.): what is the difference between these two scenarios, and what might be a reason to use an interface rather than an abstract class (and vice versa)?

In the first example, all of the objects are beverage, so we can use an abstract parent class, while in the second one, it contains drinks that are not beverage, so we may not use an abstract class.

15. List two things that you think demonstrate good design in the Ticket Vendor CRC model and two things that you think demonstrate bad design or will lead to issues when trying to implement the model in Java.
Good: 1.Building relationship between classes and methods in the design. 2.Clarify the structure of the project. Bad: 1.Team members know nothing about different part of the project assigned. 2.Cannot get into approvement in the team before starting the project.