Week 2

Java (based on course notes, Quercus quizzes, and GitHub Classroom exercises)

1. What does it mean for a method to be overloaded? What does it mean to override a method? And how does this compare to Python?
Overload means two or more methods with the same name but with different parameters.
Override means to override the method with the same name in the parent class.
We cannot override in Python.

2. What is polymorphism? Give an example of polymorphism.
Using different ways to express an action. Example paying money in int and in double creates two methods with the same name but with different parameters and data types.

3. Give two examples of casting: one of upcasting, and one of downcasting.
Dog to mammal
Mammal to dog

4. What can interfaces enforce? What can they not enforce?
Enforce: methods in the class implementing it
Not enforce: instance variables in the class

5. When would we use an interface over a parent class? How do we know which one to use?
We use an interface whenever it is possible. Otherwise use a parent class.

Git

6. This week, the coding exercise used Git with MarkUs. What were the differences between submitting to MarkUs, and submitting to GitHub?
Submitting to markus required us to contain files into the markus folder and commit, while in GitHub we can just submit in the same project cloned from GitHub.

File System (File Structure Quiz / case study / https://en.wikipedia.org/wiki/Path_(computing))

7. What is a path?
A path is a string to define where the file is stored in the directory structure.

8. How can you find the path of a given file on your computer?
Find by opening the folders instructed by the path and open the folders to search for the subfolders until we find the given file.

Case Study (lecture and case study code on GitHub)

9. List at least one example of too little or too much documentation in a class or interface.
Too little: IShellState with no documentation
10. Locate the main method in the code. Is the main method too short? Too long? What is a benefit of having fewer lines of code in main?

In Jshell. The method is neither too long and too short. The benefit is that we don’t have difficulties testing.
11. Identify an interface in the code and explain what you think its purpose is.
Interface: IShellState. Its purpose is to interact with the current state of the shell.
12. How would one add a new command to the program? Try to identify everywhere in the code where something would need to either change or be added.
Add a new command class, then add the command to the static Command_DIC in the constants.java.
13. When writing our code, we typically want to keep things as general as possible. Directory.java has an example of a too-specific return type (ArrayList). What could the return type be replaced with?
List

Test-Driven Development (lecture)

14. What are some benefits of writing tests first?
Tell other people how the program actually works.

15. What is unit testing? Why shouldn't we just test the whole program all at once?
A test used for testing every part of the program is doing well. We shouldn’t because it will cost much time and be difficult to locate mistakes.

16. What are some features of IntelliJ that help developers write / make use of junit tests during development?
Group assertions, Nested annotation, throw expectations.

Lab 3

17. What are some strategies or actions that might make working in a team easier for this course? (e.g. communication strategies, methods of code review, organization)
Assign work clearly. Understand people. Talk kindly. Be patient. Do not push all works until the deadline. Discuss before making decisions.

18. List two problems you found in your group member's code during the Trader exercise. What benefits are there to code reviews? Do you think code reviews are worth the time investment?
Lack of documentation.
Strange names for naming classes and methods.
Acknowledge what changes have been made by teammates.
Yes