Lab D Review Form

This form was completed by group: Group 23
The code reviewed was written by group: Group 1

Style

Questions

1. Are the lines short enough in length?
   Yes, the lines are short enough in length.

2. Are the methods short enough for you to comprehend the entire method at once?
   The methods are short enough.

3. Are the names of and variables and methods which were not explicitly named in the exercise sensible?
   There are no new variables and methods defined.

Scope

Questions

4. Are variables declared only within the scope in which they are being used?
   Yes, they are.

5. Should static be used in this exercise? Why or why not? If yes, where?
   Static was not needed in this exercise since it was just debugging the files.

6. Is static used appropriately?
   Yes, the group was not required to write code that uses static.

7. Should final be used in this exercise? Why or why not? If yes, where?
   The group was not required to write code involving final in this lab.

8. Is final used appropriately? If it is used, is it written according to convention (e.g., all caps)?
   Yes, the group was only required to debug the code and didn't need to use final.

9. Are access modifiers (public, private, etc) used appropriately?
   Yes, they are used correctly.

Documentation

Questions

10. Can you run `javadoc` from the command-line? If not, what was the error message?
    The Javadoc ran successfully but it gave errors for the test file which is expected because the test isn't supposed to be part of the project.

11. Does the program contain an appropriate header in the comments?
    The program does not contain any appropriate header in the comments apart from the default header given by the course instructor.

12. Is a description of the class provided?
    There is no explicit description of the class.

13. Does each method have a description?
    Each method has a valid and concise description within the classes.

14. Does each method describe `@return` and `@param` if applicable? The description should explain what is expected, not simply name a variable.
    Each method does not describe @return and @param and this may not be applicable.

15. Are the description of the class, the descriptions of the method, and the return values and parameters present in the generated documentation? If they are present in the Java file, but cannot be seen in the generated documentation, explain what mistakes were made in writing the Java documentation.
    The generated documentation did not include the description of the class, the method, and the return values and parameters. This was expected since the only things that were included were the comments.

16. Are comments used to explain the code? Do they make sense?
    Comments are used to explain the code and they do make sense.

Code Review (Reading)

Questions

17. Is the code part of the package `edu.ucalgary.oop`?
    Yes, it is part of the package.

18. Are the files named correctly?
    Yes, all files are named correctly.

19. Does the submission contain only the files which were meant to be included in the exercise?
    Yes, submission contains only the files that were meant to be included.

20. Is the submission free of extra directories and does it conform to the expected directory structure?
    Yes, the code was correctly submitted with only the required files in a folder with the group number.

21. Compare the implementation of the code to your own group's implementation. Is there anything that your group (reviewer) did more efficiently which you think the other group (reviewed) could learn from?
    No, both codes were completed very similar.

22. Is there anything that the other group (reviewed) did more efficiently than your own group (reviewer) that you learned from?
    No, both codes were completed very similar.

Code Review (Execution)

Questions

23. Did all tests pass? If any tests failed, note which tests failed and the error messages. Explain what the error means. Suggest how the authors can approach debugging their code.
    All the tests passed!

Further Understanding

Questions

24. Create a UML diagram depicting AccessLog.java and ManageAccess.java based on the code you are reviewing.