Objectives

To continue practicing the use of GitHub to access the files for a practical assignment. Additionally, to practice using the Ubuntu operating system and software development program such as a “terminal window”. You will continue to practice using Slack to support communication with the teaching assistants and the course instructor. Next, you will learn how to fix a Java program, further discovering how the course’s automated grading tool assesses your progress towards correctly completing the project. Finally, you will continue to learn more about variables and data types and the errors that may arise when variables have the wrong types.

Reading Assignment

If you have not done so already, please read all of the relevant “GitHub Guides”, available at https://guides.github.com/, that explain how to use many of the features that GitHub provides. In particular, please make sure that you have read guides such as “Mastering Markdown” and “Documenting Your Projects on GitHub”; each of them will help you to understand how to use both GitHub and GitHub Classroom. To do well on this assignment, you should also read Chapters 1 and 2 in the course textbook, ensuring that you fully understand all of the concepts that we discussed during class and investigated during prior practical and laboratory sessions. Please see the instructor or one of the teaching assistants if you have questions these reading assignments.

Finding and Fixing Defects in Programs

To access the practical assignment, you should go into the #practicals channel in our Slack team and find the announcement that provides a link for it. Copy this link and paste it into your web browser. Now, you should accept the practical assignment and see that GitHub Classroom created a new GitHub repository for you to access the assignment’s starting materials and to store the completed version of your assignment. Specifically, to access your new GitHub repository for this assignment, please click the green “Accept” button and then click the link that is prefaced with the label “Your assignment has been created here”. If you accepted the assignment and correctly followed these steps, you should have created a GitHub repository with a name like “Allegheny-Computer-Science-111-S2018/computer-science-111-spring-2018-practical-3-jjumadinova”. Unless you provide the instructor with documentation of the extenuating circumstances that you are facing, not accepting the assignment means that you automatically receive a failing grade for it. Please see the instructor if you are stuck on getting started.

Now, study the documentation in the provided source code to understand the type of output that your program should produce. Note that the provided source code contains defects in it! You are responsible for finding and fixing all of these defects. Once you locate and resolve one of these issues, please put a comment into the code to explain the problem that you found and the way in which you decided to handle it. As you complete this practical assignment, make sure that you regularly commit your code to GitHub and use descriptive messages that say what you fixed.
Checking the Correctness of Your Program

As in the past assignments, you are provided with an automated tool for checking the quality of your source code. Please note that the practical assignments do not require you to produce a writing document as you do in the laboratory assignments. However, to check your Java source code you can started with the use of GatorGrader, type the command "./gatorgrader.sh --start" in your terminal window. Once this step completes you can type "/gatorgrader.sh --check". If your work does not meet all of the assignment’s requirements, then you will see the following output in your terminal: “Overall, are there any mistakes in the assignment? Yes”. If you do have mistakes in your assignment, then you will need to review GatorGrader’s output, find the mistake, and try to fix it. Remember, this practical assignment provides you with Java source code that purposefully contains mistakes — your task is to find and fix these problems!

Once your program is building correctly, fulfilling at least some of the assignment’s requirements, you should transfer your files to GitHub using the “git commit” and “git push” commands. For example, if you want to signal that the src/main/java/practicalthree/ComputeMoonDistance.java file has been changed and is ready for transfer to GitHub you would first type “git commit src/main/java/practicalthree/ComputeMoonDistance.java -m "Your descriptive commit message"” in your terminal, followed by typing “git push” and checking to see that the transfer to GitHub is successful. If you notice that transferring your code or writing to GitHub did not work correctly, then please try to determine why, asking a teaching assistant or the course instructor for help, if necessary.

After the course instructor enables “continuous integration” with a system called Travis CI, when you use the “git push” command to transfer your source code to your GitHub repository, Travis CI will initialize a “build” of your assignment, checking to see if it meets all of the requirements. If both your source code and writing meet all of the established requirements, then you will see a green ✔ in the listing of commits in GitHub after awhile. If your submission does not meet the requirements, a red ✕ will appear instead. You should aim to finish practical assignments on the day that they are assigned; please see the instructor if you do not understand this policy.

Summary of the Required Deliverables

This assignment invites you to submit, using GitHub, the following deliverables. Because this is a practical assignment, you are not required to complete any technical writing.

1. A properly documented, well-formatted, and correct version of src/main/java/practicalthree/ComputeMoonDistance.java that both meets all of the established requirements and produces the desired output. In addition to your Java source code no longer containing any defects that “break the build”, it must feature comments that explain each of the mistakes that you found and a clear statement of how you fixed them.

Evaluation of Your Practical Assignment

Practical assignments are graded on a completion — or “checkmark” — basis. If your GitHub repository has a ✔ for the last commit before the deadline then you have completed the assignment to the highest possible level. In order to receive a “checkmark” for this assignment you need to commit and push your work (even if incomplete) before the stated deadline. Please see the course instructor if you do not understand how practical assignments are graded or you do not know how to complete one of the specific tasks in this assignment.