Objectives

To practice using 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 also continue to practice using Slack to support communication with the teaching assistants and the course instructor. Next, you will learn how to implement a Java program and also discover how to use the course’s automated grading tool to assess your progress towards correctly completing the project. Finally, you will explore how to perform character output and to use escaped characters.

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, paying particularly close attention to Sections 1.5 and 2.1. Please see the instructor or one of the teaching assistants if you have questions these reading assignments.

Display Character-Based Artwork

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-2-jjamadinova”. 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 follow the steps from the previous laboratory assignment for this practical assignment; 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 the constraints on the character symbols that should appear in the output and the number of possible println statements you can use to produce the output.

Checking the Correctness of Your Program

As in the previous assignment, 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. 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/practicaltwo/DisplayArtwork.java file has been changed and is ready for transfer to GitHub you would first type “git commit src/main/java/practicaltwo/DisplayArtwork.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 to GitHub did not work correctly, then please try to determine why, asking an assistant or the instructor for assistance, 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 your source code meets 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. The instructor will reduce a student’s grade for this assignment if the red ✗ appears on the last commit in GitHub immediately before the assignment’s due date. Yet, if the green ✓ appears on the last commit in your GitHub repository, then you satisfied all of the main checks. Unless you provide the course instructor with documentation of the extenuating circumstances that you are facing, no late work will be considered towards your completion grade for this practical assignment.

Summary of the Required Deliverables

Students do not need to submit printed source code or technical writing for any assignment in this course. Instead, 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/practicaltwo/DisplayArtwork.java that both meets all of the established requirements and produces the desired output.

Adhering to the Honor Code

In adherence to the Honor Code, students should complete this assignment on an individual basis. While it is appropriate for students in this class to have high-level conversations about the assignment, it is necessary to distinguish carefully between the student who discusses the principles underlying a problem with others and the student who produces assignments that are identical to, or merely variations on, someone else’s work. Deliverables (e.g., Java source code or Markdown-based technical writing) that are nearly identical to the work of others will be taken as evidence of violating the Honor Code. Please see the course instructor if you have questions about this policy.