Objectives

To continue practicing the use of GitHub to access the files for a practical assignment. You will also continue to practice using Slack to support communication with the teaching assistants and the course instructor. Next, you will learn more about random number generation and the use of `java.util.Random` and `java.lang.Math`, further discovering how the course’s automated grading tool assesses your progress towards correctly completing the project.

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. Focusing on the content about creating and using Java objects, you should review Chapters 1 through 3 in the textbook.

Using Random and Math Classes

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. Please follow the steps from the previous assignments to accept the assignment, create an assignment repository, and to download it locally.

Figure 1 contains the output from running a program like the one you must implement. Specifically, your program should perform the following steps:

1. Declare two `int` variables to store face value of each die.
2. Simulate each die roll. For this step you can utilize either `Random` or `Math` class’s method to generate a random number in an appropriate range. Please note that if you use `Math` class’s method to generate a random number you will need to ensure that the result is of type `int`.
3. Print out the face value of each die.
4. Compute and output the sum of each die’s face value.
5. Take the first die’s value to the power of the second die’s value and output the result. Use an appropriate method from the `Math` class.
6. Find and output the maximum of the two dice values. Use an appropriate method from the `Math` class.
7. Find and output the minimum of the two dice values. Use an appropriate method from the `Math` class.
> Task :run
> Die One’s Face Value: 2
> Die Two’s Face Value: 5
> The Sum of the Two Die is: 7
> The power result is: 32.0
> The maximum is: 5
> The minimum is: 2

Figure 1: Sample “RollDice” output featuring randomly generated values.

After finishing the src/main/java/practicalsix/RollDice.java file, you should repeatedly test you program to make sure that it is creating the correct textual output. This will involve you building and running the program and checking the output to ensure that it produces different values and that the mathematical calculations are correct. You must also have correct formatting and documentation in your program.

Checking the Correctness of Your Program and Writing

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 “gradle grade” in your terminal window. 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. Specifically, don’t forget to add in the required comments! If you are having trouble running GatorGrader locally, don’t forget to ensure that you still transfer all of your source code to GitHub. Please see the course instructor if you have questions about this step.

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/practicalsix/RollDice.java file has been changed and is ready for transfer to GitHub you would first type “git commit src/main/java/practicalsix/RollAndSumDice.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 a teaching assistant or the course instructor for help, if necessary.

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 correct version of src/main/java/practicalsix/RollDice.java that meets all of the established source code requirements and produces the desired text-based output.
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 will receive the highest possible grade for the assignment. However, you will fail the assignment if you do not commit and push your program, by the set deadline for completing the project. 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.

Adhering to the Honor Code

In adherence to the Honor Code, students should complete this practical 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., the Java source code) 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.