Objectives

In this practical assignment, you will explore an already implemented Java program that causes various types of exceptions to be thrown during a program run. As you modify the program, you will uncomment certain sections of it and then observe, record, and comment on the output that is produced. The goal for this assignment is to ensure that you understand how the throwing of an exception influences the execution of a Java program. Finally, you will continue to practice writing technical documentation in Markdown and using a Git repository hosted by GitHub.

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 GitHub’s features. 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 learn more about the concepts associated with exception handling in the Java programming language, please study the content in Chapter 11. In particular, make sure that you understand the purpose of exceptions, the messages displayed in the terminal when an exception is uncaught, and the structure and meaning of the try-catch-finally blocks.

Using and Extending a Program that Throws Exceptions

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.

After reviewing the provided source code file, please use Gradle to build and run the src/main/java/practicalten/InvestigateExceptions.java program. At the outset, you will see that this program does not throw any exceptions at all. So, what you should do next is to incrementally and individually uncomment each of the calls to the throwsExceptions method and observe and understand the output from the program. The idea is that you will uncomment a method call in main, build and run the program, observe and understand the program’s output, comment out that line again, and then move onto the next line of code in the program’s main method.

Each time the program throws a different exception, make sure that you understand why it does so. Next, you will notice that, in certain cases, the program does not output a full “stack trace” that prints out in the terminal window which exception is thrown. As such, you may want to add code to the catch blocks of ExceptionExample that can print the stack trace. Finally, you must create a file called writing/exceptions.md that explains the output from your various runs of the program. Please write about what exception was thrown and explain why it was thrown.
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 "/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. Specifically, don’t forget to add in the required writing! 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/practicalten/InvestigateExceptions.java file has been changed and is ready for transfer to GitHub you would first type “git commit src/main/java/practicalten/InvestigateExceptions.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.

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. Since this is another challenging practical assignment and you are continuing to learn how to throw exceptions, don’t become frustrated if you make a mistake. Instead, use your mistakes as an opportunity for learning both about the necessary technology and the background and expertise of the other students in the class, the teaching assistants, and the course instructor.

This assignment invites you to submit, using GitHub, the following deliverables.

1. A completed version of the writing/exceptions.md file that reports on the output and behavior of the provided program when it individually triggers each exception.

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. 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. Finally, remember that, in adherence to the Honor Code, students should complete this practical assignment on an individual basis. Please see the course instructor if you have any questions about this course policy.