Objectives

To continue to practice using GitHub to access the files for a laboratory assignment. Additionally, to master the use of the Ubuntu operating system and software development programs 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 that creates graphical objects and displays them on the user’s screen. Finally, you will continue to explore the use of methods, variables, and data types.

Suggestions for Success

- **Use the laboratory computers.** The computers in this laboratory feature specialized software for completing this course’s laboratory and practical assignments. If it is necessary for you to work on a different machine, be sure to regularly transfer your work to a laboratory machine so that you can check its correctness. If you cannot use a laboratory computer and you need help with the configuration of your own laptop, then please carefully explain its setup to a teaching assistant or the course instructor when you are asking questions.

- **Follow each step carefully.** Carefully read each sentence in the assignment sheet, making sure that you precisely follow each instruction. Take notes about each step that you attempt, recording your questions and ideas and the challenges that you faced. If you are stuck, then please tell a teaching assistant or instructor what assignment step you recently completed.

- **Regularly ask and answer questions.** Please log into Slack at the start of a laboratory or practical session and then join the appropriate channel. If you have a question about one of the steps in an assignment, then you can post it to the designated channel. Or, you can ask a student sitting next to you or talk with a teaching assistant or the course instructor.

- **Store your files in GitHub.** As in the past laboratory assignments, you will be responsible for storing all of your files (e.g., Java source code and Markdown-based writing) in a Git repository using GitHub Classroom. Please verify that you have saved your source code in your Git repository by using “git status” to ensure that everything is updated. You can see if your assignment submission meets the established correctness requirements by using the provided checking tools on your local computer and in checking the commits in GitHub.

- **Keep all of your files.** Don’t delete your programs, output files, and written reports after you submit them through GitHub; you will need them again when you study for the quizzes and examinations and work on the other laboratory, practical, and final project assignments.

- **Back up your files regularly.** All of your files are regularly backed-up to the servers in the Department of Computer Science and, if you commit your files regularly, stored on GitHub. However, you may want to use a flash drive, Google Drive, or your favorite backup method to keep an extra copy of your files on reserve. In the event of any type of system failure, you are responsible for ensuring that you have access to a recent backup copy of all your files.
• Explore teamwork and technologies. While certain aspects of the laboratory assignments will be challenging for you, each part is designed to give you the opportunity to learn both fundamental concepts in the field of computer science and explore advanced technologies that are commonly employed at a wide variety of companies. To explore and develop new ideas, you should regularly communicate with your team and/or the teaching assistants and tutors.

• Hone your technical writing skills. Computer science assignments require you to write technical documentation and descriptions of your experiences when completing each task. Take extra care to ensure that your writing is interesting and both grammatically and technically correct, remembering that computer scientists must effectively communicate and collaborate with their team members and the tutors, teaching assistants, and course instructor.

• Review the Honor Code on the syllabus. While you may discuss your assignments with others, copying source code or writing is a violation of Allegheny College’s Honor Code.

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, you should 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 review Chapter 1 and study Chapter 2 in the course textbook, paying close attention to Sections 2.7 through 2.9 and Figures 2.9, 2.10, and 2.12. See the instructor if you have questions these reading assignments.

Accessing the Laboratory Assignment on GitHub

To access the laboratory assignment, you should go into the #labs 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 laboratory 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-lab-3-jjumadinova”. If you provided 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.

Before you move to the next step of this assignment, please make sure that you read all of the content on the web site for your new GitHub repository, paying close attention to the technical details about the commands that you will type and the output that your program must produce. Now you are ready to download the starting materials to your laboratory computer. Click the “Clone or download” button and, after ensuring that you have selected “Clone with SSH”, please copy this command to your clipboard. In your terminal navigate to the directory where your lab repositories are stored. Now by typing “git clone” in your terminal and then pasting in the string that you copied from the GitHub site you will download all of the code for this assignment. For instance, if the course instructor ran the “git clone” command in the terminal, it would look like:

```
$ git clone git@github.com:Allegheny-Computer-Science-111-S2018/computer-science-111-spring-2018-lab-3-jjumadinova.git
```

After this command finishes, you can use “cd” to change into the new directory. If you want to “go back” one directory from your current location, then you can type the command “cd ..”. Please
continue to use the “cd” and “ls” commands to explore the files that you automatically downloaded from GitHub. What files and directories do you see? What do you think is their purpose? Spend some time exploring, sharing your discoveries with a neighbor and a teaching assistant.

**Creating Sophisticated Graphical Output**

To load the source code of today’s Java program into atom, you should be inside your newly created repository (e.g., “computer-science-111-spring-2018-lab-3-jjumadinova”). Then you can type “atom src/main/java/labthree/DisplayGraphicalScene.java” in your terminal window. This should cause a new window, the atom text editor, to appear on your screen with a partially completed program.

This assignment invites you to implement desktop wallpaper art similar to the examples that are found on the web site called http://simpledesktops.com/. First, your Java program should display your name and then the date on which the program was run. You can accomplish this task with “JFrame window = new JFrame(“Janyl Jumadinova " + new Date());”, substituting your name for the name of the instructor. Please see the Java source code comments for more details about the program’s purpose! The remainder of this section summarizes the goals for this program.

First, this assignment asks you to write a program, src/main/java/labthree/DisplayGraphicalScene.java, that will display a canvas for a graphical scene. In particular, it must do the following:

1. Contain at least four single-line comments and two multi-line comments.
2. Declare and use the “CANVAS_HEIGHT” variable.
3. Declare and use the “CANVAS_WIDTH” variable.

Second, this assignment invites you to write another program, called src/main/java/labthree/PaintGraphicalScene.java, that will paint a graphical scene on an already displayed canvas.

1. Contain at least four single-line comments and two multi-line comments.
2. Declare and use the “CANVAS_HEIGHT” variable.
3. Declare and use the “CANVAS_WIDTH” variable.
4. Create at least five colors using the “page.setColor” method call.
5. Create at least three rectangles using the “fillRect” method call.
6. Create at least five ovals using the “fillOval” method call.
7. Declare and use at least two “int” variable when creating and displaying graphics.

If you want to “build” your program you can type the command “gradle build” in your terminal, thereby causing the Java compiler to check your program for errors and get it ready to run. If you get any error messages, go back into atom and try to figure out what you mis-typed and fix it. Once you have solved the problem, make a note of the error and the solution for resolving it. Re-save your program and then build it again by re-running the “gradle build”. If you cannot build DisplayGraphicalScene correctly, then please talk with a teaching assistant or the instructor.

When all of the errors are eliminated, you can run your program by typing “gradle run” in the terminal window—this is the “execute” step that will run your program and produce the designated output. You should see your name, today’s date, and the graphical output. Make sure there are spaces separating words in your output (did you forget to put a space inside the quotation marks after your last name?). If not, then repair the program and re-build and re-run it. Once the program runs, please reflect on this process. What step did you find to be the most challenging? Why? You should write your reflections in a file, called writing/reflection.md, that uses the Markdown writing language. To complete this aspect of the assignment, you should write two high-quality paragraphs that describe the artistic expression you decided to pursue and also report on your experiences with the various commands and Java code segments.

Due: Section 1: 15 February, Section 2: 14 February, 2018 at 2:30 pm
Checking the Correctness of Your Program and Writing

As verified by Checkstyle, the code for the src/main/java/labthree/DisplayGraphicalScene.java and the src/main/java/labthree/PaintGraphicalScene.java files must adhere to all of the requirements in the Google Java Style Guide available at https://google.github.io/styleguide/javaguide.html. The Markdown file that contains your reflection must adhere to the standards described in the Markdown Syntax Guide https://guides.github.com/features/mastering-markdown/. Instead of requiring you to manually check that your deliverables adhere to these industry-accepted standards, the GatorGrader tool that you will use in this laboratory assignment makes it easy for you to automatically check if your submission meets these well-established standards for correctness.

Since this is not your first laboratory assignment, you will notice that the provided source code does not contain all of the required comments at the top of the Java source code file. This means that you will have to inspect the source code from previous laboratory and practical assignments to review how to create the comments in the src/main/java/labthree/DisplayGraphicalScene.java and src/main/java/labthree/PaintGraphicalScene.java files. Moreover, the provided source code is missing many of the lines that are needed to pass the GatorGrader checks. Please review the requirements for these two Java source code files, as outlined in the previous section. You should also study the source code of these files to learn more about what you need to add to them.

To get 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/labthree/DisplayGraphicalScene.java file has been changed and is ready for transfer to GitHub you would first type “git commit src/main/java/labthree/DisplayGraphicalScene.java -m "Your descriptive commit message"” in your terminal, followed by typing “git push”, and then checking to see that the transfer to GitHub is successful. Remember, to correctly complete this assignment you need to commit all code and writing files to GitHub. If you notice that the network communication with GitHub did not work, then please try to determine why, asking a teaching assistant or the instructor for assistance.

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. 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, thereby allowing the course instructor to evaluate other aspects of your source code and writing, as further described in the “Evaluation” section of this assignment sheet. In conclusion, here are some points to remember for creating programs that display graphical output:

1. You should think carefully about how your graphics can be displayed using variables.
2. As in past assignments, your program only needs to have one main method in one file.
3. See Figure 2.10 for a listing of some common “Color” variables for use in your program.
4. Your program will alternate between creating and displaying graphical output—this is okay!
5. Don’t forget to review the assignment sheets from the previous laboratory and practical assignments as they contain insights that will support your completion of this assignment.

**Summary of the Required Deliverables**

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

1. Stored in `writing/reflection.md`, a two-paragraph reflection on the graphical output you decided to create and the details of your implementation. For example, this Markdown-based document should explain what your graphical output expresses, your thoughts behind the process of creating your graphical output, the input, output, and behavior of the Java commands you have utilized and the challenges that you confronted when using it. For every challenge that you encountered, please explain your solution for it.

2. A complete and correct version of `src/main/java/labthree/DisplayGraphicalScene.java` that both meets all of the established requirements and produces the desired output.

3. A complete and correct version of `src/main/java/labthree/PaintGraphicalScene.java` that both meets all of the established requirements and produces the desired output.

**Evaluation of Your Laboratory Assignment**

Using a report that the instructor shares with you through the commit log in GitHub, you will privately received a grade on this assignment and feedback on your submitted deliverables. Your grade for the assignment will be a function of the whether or not it was submitted in a timely fashion and if your program received a green ✓ indicating that it met all of the requirements. Other factors will also influence your final grade on the assignment. In addition to studying the efficiency and effectiveness of your Java source code, the instructor will also evaluate the accuracy of both your technical writing and the comments in your source code. If your submission receives a red ✗, the instructor will reduce your grade for the assignment while still considering the regularity with which you committed to your GitHub repository and the overall quality of your partially completed work. Please see the instructor if you have questions about the evaluation of this laboratory assignment.