

CMPSC 111
Introduction to Computer Science I
Fall 2015
Lab 5
23 September 2015
Due Wednesday, 30 September by 2:30pm

Objectives

In addition to enhancing the skills that you have learned in the past laboratory assignments, the purpose of this assignment is to explore the ideas of a “class” and an “object” in the Java programming language. Also, you will learn how to use the methods provided by the `java.lang.String` class to inspect and manipulate a `String` object. You will then apply your knowledge of `Strings` to the application domain of steganography, or the practice of “hiding” messages inside of non-secret content. Finally, you will work in a team to complete a real-world programming task, using tools like Git and Slack to support your collaborative completion of the project.

General Guidelines for Labs

- **Work on the Alden Hall computers.** If you want to work on a different machine, be sure to transfer your programs to the Alden machines and re-run them before submitting.
- **Update your repository often!** You should add, commit, and push your updated files each time you work on them. I will not grade your programs until the due date has passed.
- **Review the Honor Code policy.** You may discuss programs with others, but programs that are nearly identical to others will be taken as evidence of violating the Honor Code.

Reading Assignment

To review what you have already learned about variables, expressions, and user input, please read Sections 2.1–2.6 in your textbook; pay close attention to the `Scanner` methods in Figure 2.7 and the program in Listing 2.8. To learn more about Java classes and objects and, in particular, the methods provided by the `java.lang.String` class, please study Sections 3.1–3.2 in the textbook. Finally, don’t forget to examine the “Git Cheatsheet” if you have questions about how to create and use a new Git repository. Please see the instructor if you have questions about these readings.

Creating Your Team’s Collaboration Methods

For this assignment, please pick a partner—in particular, an individual with whom you have not already worked on an in-class assignment. You and your partner are responsible for implementing a complete solution to a real-world problem involving the storage and transmission of secret messages using steganography. To start, you and your partner should create a new Git repository hosted by Bitbucket. This new repository must adhere to the naming convention `<first-partner-username>-<second-partner-username>-cs111F2015-lab5`. After creating the repository with the correct name, please share it with each other and with the course

instructor. Now, make sure that you can both access this repository through Git commands such as `clone`, `pull`, `add`, `commit`, and `push`. Additionally, you should make your own `lab5/` directory in your personal repository and make sure that, by the end of the laboratory assignment, all of the files that you collaboratively created are also available in this directory. Please see the course instructor or one of the teaching assistants if you cannot complete these first two steps.

Before you move on to the next phase of the assignment, one member of your team should also create a Slack channel that can host all of the communication between you and your partner as you complete this assignment. Please make sure that you both are a member of the channel and that you add the course instructor as well (remember, my user name in Slack is “`gkapfham`” and you can best ensure that I read and respond to your question by using the notation “`@gkapfham`”).

Collaboratively Implementing a Steganography Program

Within the field of computer security, there are many sub-fields that develop strategies for sending, receiving, and storing secret messages. Involving the hiding of a secret “in plain sight”, steganography is one way in which you can create and send a secret message. For this assignment, you and your partner will implement a program called `WordHide.java` that will perform these operations:

1. Prompt the user for a word that is exactly ten characters in length. For now, `WordHide` should allow the user to enter any type of “padding character” before or after a word that is less than ten characters. If the word is longer than ten characters, then your program should simply discard all of the characters after the tenth one (you may also implement better strategies).
2. Since the `WordHide` program must output all of its character in a capitalized form, you should transform the word provided by the user so that it only contains upper-case letters.
3. Finally, `WordHide` should output a 20×20 “grid” of letters that contain the user’s word “hidden” inside of it. All of the letters in this grid should be capitalized. You and your partner should brainstorm and prototype different techniques for effectively hiding the user’s word in the grid of letters. As you implement your program, you must make decisions about the following matters: (i) what letters will you add to the grid to best hide the user’s word? (ii) where will you place the user’s word in the grid? (iii) what features of the Java programming language will you use to ensure the grid is formatted properly in the terminal? As you answer these questions and finish implementing `WordHide.java`, it may help to consider the fact that the user’s word might be better hidden if the grid contains letters found in the input.

Carefully Review the Honor Code

The Academic Honor Program that governs the entire academic program at Allegheny College is described in the Allegheny Academic Bulletin. The Honor Program applies to all work that is submitted for academic credit or to meet non-credit requirements for graduation at Allegheny College. This includes all work assigned for this class (e.g., examinations, laboratory assignments, and the final project). All students who have enrolled in the College will work under the Honor Program. Each student who has matriculated at the College has acknowledged the following pledge:

I hereby recognize and pledge to fulfill my responsibilities, as defined in the Honor Code, and to maintain the integrity of both myself and the College community as a whole.

It is understood that an important part of the learning process in any course, and particularly one in computer science, derives from thoughtful discussions with teachers and fellow students. Such dialogue is encouraged. However, 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. While it is acceptable for students in this class to discuss their programs, data sets, and reports with their classmates, deliverables that are nearly identical to the work of others will be taken as evidence of violating the Honor Code. For this assignment, it is crucial that you only share source code segments with your chosen partner!

Required Deliverables

In addition to submitting signed and printed versions of all materials, for this assignment you are invited to submit electronic versions of the following deliverables through the Bitbucket repository. As you complete this step, you should make sure that you created a `lab5/` directory within your own Git repository. Then, you can save all of the required deliverables in the `lab5/` directory—please see the course instructor or a teaching assistant if you are not able to create your directory properly. Additionally, you and your partner should have created a separate repository to support the completion of this assignment; it should also contain a `lab5/` directory with all of the deliverables that you collaboratively created for this assignment. Finally, please make sure that your team creates a Slack channel that you can use to support all of your communication for this assignment.

1. A completed, properly commented and formatted `WordHide.java` program. Please make sure that your Java program includes the comment header file with the Honor Code, both member’s name, the date, and the description of the program—in addition including code for displaying the laboratory assignment’s number and the date on which the program was run.
2. Three outputs from running `WordHide` in the terminal window three times with three different inputs from the user. Your output should clearly demonstrate how your program hides the user’s input somewhere inside of the 20×20 grid of characters. You may use `gvim` to save all three of your outputs as follows: using the mouse, select everything from the “`java WordHide`” command to the end of your output. Right-click on the selected text and copy it. Type “`gvim output`”—note that this *not* a Java program!—and use the “Edit/Paste” menu item to paste your program’s output into the file. Now, use “`:w`” or the “File/Save” menu item to save this file. Please see the course instructor if you cannot save your output file.
3. A document called `team_roles` that describes, in detail, the tasks that each member of your team completed. For each member, you should state the description of and due date for their task(s) and whether or not they actually succeeded in completing the task by the deadline.

Share your program and the output file with me through your Git repository by correctly using “`git add`”, “`git commit`”, and “`git push`” commands. When you are done, please ensure that the Bitbucket Web site has a `lab5/` directory in your repository with only the three files called `WordHide.java`, `output`, and `team_roles`. Remember, you should ultimately have the three files for this laboratory assignment in both your own personal Git repository and the special repository that you created specifically for the completion of this assignment. You should review the “Git Cheatsheet” and see the instructor if you have questions about assignment submission with `git`.