

CMPSC 111
Introduction to Computer Science I
Fall 2014

Practical 3
18–19 September 2014
Due in Bitbucket by midnight of the day of your practical
“Checkmark” grade

Summary

As a means of practicing with user input and output, the declaration of variables, and the creation of expressions in the Java programming language, you will write a “Mad Libs” program involving words, numbers, and some calculations. Then, using the `git add`, `git commit`, and `git push` commands you should upload it to your Git repository hosted by Bitbucket.

If you have never heard of Mad Libs, then please visit this Web site: <http://www.madlibs.com/>.

Review the Textbook

Be sure to read Sections 3.1 through 3.6 of your book to learn more about variables, data types, expressions, and user input. As you review this material, try to make a list of questions about concepts that you do not yet fully understand. In addition to discussing these questions with the teaching assistants and a course instructor, try to answer them as you complete this assignment.

Exercise: Implement a “Mad Libs” Program

Your program should ask the user to enter words and numbers, then print out a story using those words and numbers. The numbers should be used to calculate something which will also be printed out. See Figure 1 for an example—but, of course, you should create your own story!

Remember to “`import java.util.Scanner`” at the top of your Java program and to declare a `Scanner` variable (named, for instance, `scan`—but you can name it something else if you want). Please use statements like “`... = scan.next()`”, “`... = scan.nextInt()`”, and “`... = scan.nextDouble()`” to read in strings, integers, and double values. Your program should produce output that is neat and your program’s source code must make good use of white space and labeling.

General Guidelines for Practical Sessions

- **Experiment!** Practical sessions are for learning by doing without the pressure of grades or “right/wrong” answers. So try things! The best way to learn is by trying things out.
- **Submit *Something*.** Your grade for this assignment is a “checkmark” indicating whether you did or did not complete the work and submit something to the Bitbucket repository using the “`git add`”, “`git commit`”, and “`git push`” commands.
- **Practice Key Laboratory Skills.** As you are completing this assignment, practice using the `gvim` text editor and the Ubuntu terminal until you can easily use their most important features. Additionally, ask a teaching assistant or a course instructor to teach you some of the advanced features of `gvim` and the terminal, thereby helping you to work more effectively.

```

aldenv27:gkapfham$ javac MadLib.java
aldenv27:gkapfham$ java MadLib
Gregory M. Kapfhammer, Practical 3
Thu Sep 17 13:11:17 EST 2014

Enter a singular noun: noggin
Enter an adjective: verboten
Enter another adjective: glitzy
Enter a non-zero whole number: 32
Enter another non-zero whole number: 42
Enter any number: 5.43
Enter a singular verb: snooze

```

Third-Grade Word Problem

If you own 32 verboten noggins,
and you purchase 42 glitzy noggins,
how many more noggins do you need to snooze 5.43 noggins?

Answer: You need -68.57 more noggins.

Figure 1: Sample “Mad Libs” output with user input highlighted in red.

- **Try to Finish During the Class Session.** Practical exercises are not intended to be the equal of the laboratory assignments. If you are simply a slow typist, I’ve given you until the end of the day, but ideally you should upload a file, even a non-working one, by the end of the class period. You also should ensure that, for this assignment and all subsequent assignments, you can confidently upload files to your Git repository during the practical session.
- **Help One Another!** If your neighbor is struggling and you know what to do, offer your help. Don’t “do the work” for them, but advise them on what to type or how to handle things. If you are stuck on a part of this practical session and you could not find any insights in either your textbook or online sources, formulate an intelligent question to ask your neighbor, a teaching assistant, or a course instructor. Try to strike the right balance between asking for help when you cannot solve a problem and working independently to find a solution.
- **Update Your Repository Often!** You should `add`, `commit`, and `push` your updated files each time you work on them, always including descriptive messages about each code change.
- **Review the Honor Code Policy on the Syllabus.** Remember that while you may discuss programs with other students in the course, programs that are nearly identical to, or merely variations on, the work of others will be taken as evidence of violating the Honor Code.