CMPSC 111
Introduction to Computer Science I
Fall 2014

Lab 1
Assigned: Thursday, 4 September 2014
Due: Thursday, 11 September 2014, 2:30 pm

Objective

To learn how to get around in the Ubuntu Linux operating system and how to create, compile, and execute simple Java programs using the powerful "`gvim`" text editor and the Ubuntu terminal.

General Guidelines

- **Use 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. However, please remember that, as stated in the syllabus, students should complete assignments using the specialized workstations in the Alden Hall laboratories; the course instructor and the teaching assistants normally are not available to help students configure their own computers.
- **Keep all of your files!** Don't delete your programs and reports after you hand them in—you will need them again later when you study for the quizzes and examinations and work on the other laboratory, practical, and final project assignments.
- **Back up your files regularly**. Use a flash drive, Google Drive, or your favorite backup method to keep a copy of your files in reserve. In the event of a system failure, you are responsible for ensuring that you have access to a recent backup copy of all their files.
- **Review the Honor Code policy on the syllabus.** Remember that you may discuss programs with others, but copying programs is a violation of the College's Honor Code.

Your Account in Alden Hall

In advance of today's lab you have already received the details about your Alden Hall computer account and learned how to log on, change your password, and log out. You should ensure that you have recorded how to complete these steps in your notebook; please report any problems as soon as they occur. You may use this account on any computer in Alden labs 101, 103, or 109. Your files are stored on a central server; you don't have to use the same machine every time you log on.

Hours of lab availability are posted on the bulletin board in each lab and on the following Web site: `http://www.cs.allegheny.edu/labs/`; the on duty lab monitor is available in Alden 101.

Creating Your First Java Program

In order to create a program, you need a text editor. There are many different text editors on your workstation and you should feel free to explore these on your own. Since it is a powerful text editor known for helping computer scientists "edit text at the speed of thought", in this class we will use the text editor called "`gvim`". Today, you will write your first program in `gvim`.

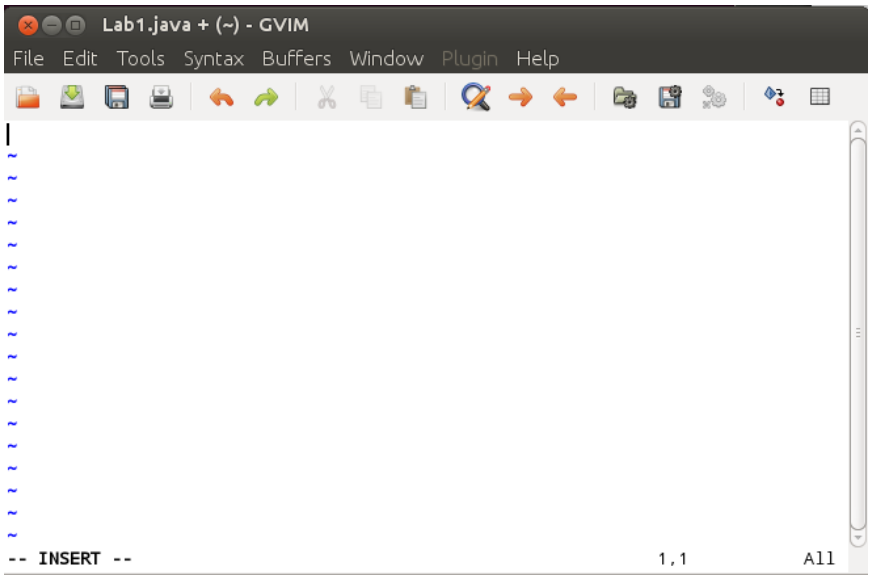

Figure 1: Insert mode in `gvim`.

There are several ways to launch `gvim`, but for today please use a method described in this paragraph. On the left side of your screen, click on the icon that contains the "`>`" symbol. Alternatively, you can type the "Super" key, start typing the word "terminal", and then select that program. Another way to open a terminal involves typing the key combination `<Ctrl>-<Alt>-t`.

Now, you should type the following commands—exactly as shown—into the terminal window. (The "`1`" is the digit "one", not the letter "ell.") All Linux, Java, and `gvim` commands are case-sensitive, so be sure to capitalize the file name "`Lab1.java`" but nothing else. Don't worry if you make a mistake—just ask the course instructor or a teaching assistant for help and then try again.

```
mkdir cs111F2014
cd cs111F2014
mkdir labs
cd labs
mkdir lab1
cd lab1
gvim Lab1.java
```

Once you have finished typing these commands and you are sure that they worked correctly, please reflect on what these steps accomplished and then make a few notes about the process. Once you think that you understand the process completely, please turn to a person sitting near you and explain it verbally. After discussing these steps with you neighbor, did you both arrive at the same understanding of their purpose? If you did not, then please talk with a teaching assistant.

After typing "`gvim Lab1.java`", a new window should appear. This is the `gvim` editor. Since `gvim` is a modal editor, you may notice that if you start typing, nothing appears (unless you happen to hit certain letters such as "`i`," "`o`", "`a`", and a few others). This is because you are not in "insert mode." To get into insert mode, just type the letter "`i`" (lower case). Once you do this, the window should look like the one in Figure 1. Note the word "`--INSERT--`" in the lower left corner!

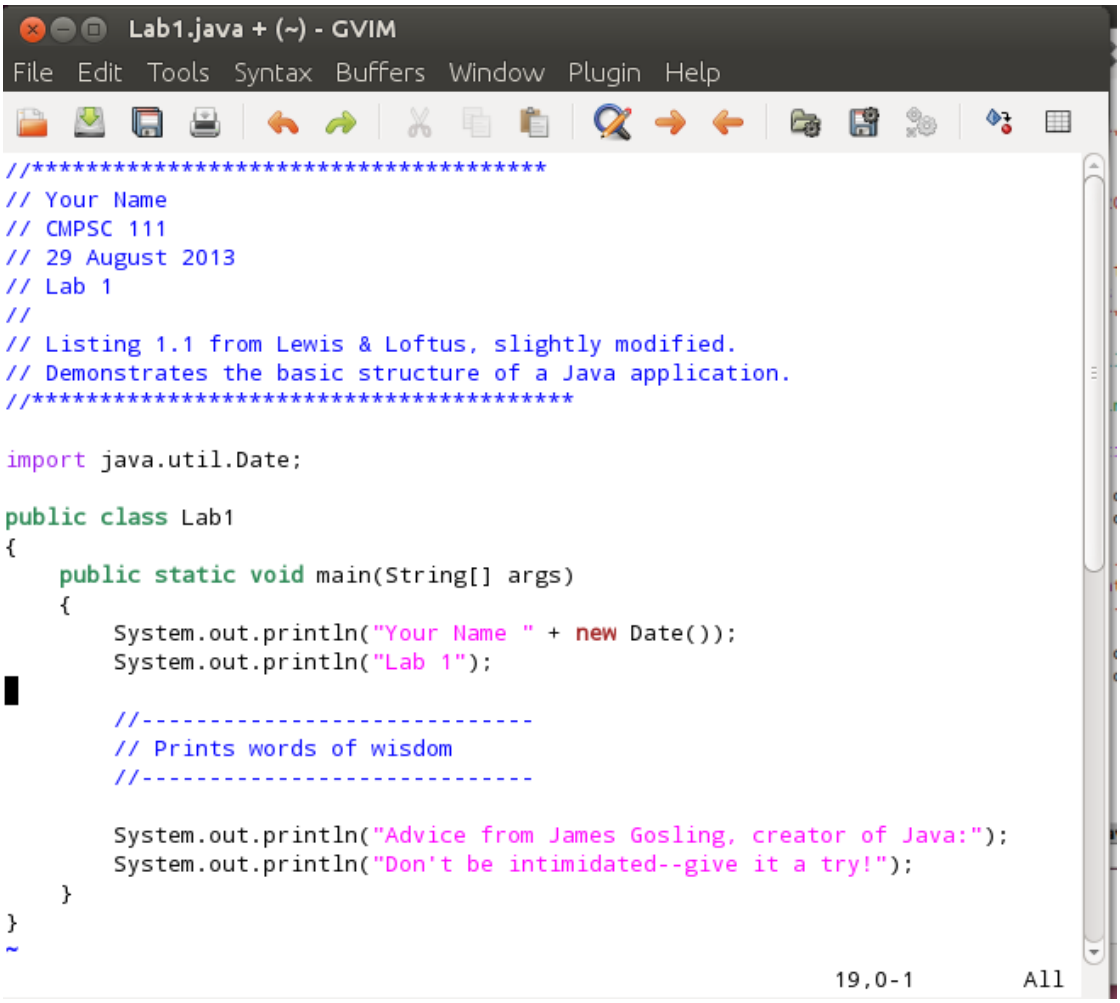

Figure 2: Your first program that outputs a quotation from James Gosling.

Type the program from Figure 2 into the window, substituting your actual name for the words "Your Name" and including today's date in place of that which is listed on the fourth line. When you are finished, press the ESC key located in the upper left corner of the keyboard. This should remove the word "`--INSERT--`" from the bottom of the screen and take you out of insert mode.

Use the "File/Save" command to save your program. Alternatively, if you would like to use the keyboard to save your file, you can press ":w" when you are not in insert mode.

Leaving the `gvim` window open, go back to your terminal window. Type the command "`javac Lab1.java`" at the terminal window's prompt—this is the "compile" step of writing a Java program that transforms the source code into a form that can be run on your computer.

If you get any error messages, go back into `gvim` 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 re-compile it (i.e., re-run the "`javac`" command). If you cannot get the program to compile correctly, then please talk with the course instructor.

When all errors are eliminated, type "`java Lab1`" 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, the lab number, and two more lines of text. 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, repair the program and re-compile and re-run it. Once the program runs, please reflect on this process. What step did you find to be the most challenging? Why?

Using the "File/Print" menu item, print out your program directly from `gvim`. Pick up your output at one of the two printers in the front of the lab (i.e., 101a or 101b). Please see the course instructor if you have trouble printing. Sign your name at the top of your printout—this is the pledge that the work you are handing in was done according to the Honor Code guidelines.

Write Your Own Program

Following the steps from the previous part of this laboratory assignment, create a program with a different name (e.g., "`Lab1Part2.java`"). Note that the name in your "`public class`" statement must exactly match the portion of the file name preceding the "`.java`". For instance, you must have "`public class Lab1Part2`" if the file name is "`Lab1Part2.java`".

Make sure that your program outputs something different than the program that you wrote previously—but still include your name and the date as shown in the example. Experiment with creating output that includes quotations, art work, or technical diagrams. At minimum, this program should create more lines of output than your first one. Try to purposefully include errors in your program by, for instance, omitting the "`;`", capitalizing something incorrectly, misspelling a Java keyword, or making other mistakes. As you learn by trying new things, ask additional questions of or give a status update to the course instructor and a teaching assistant.

Summary of the Required Deliverables

This assignment invites you to submit printed and signed versions of the following deliverables:

1. A commentary on the meaning and purpose of all the commands you typed in the terminal.
2. A properly commented and formatted version of the `Lab1.java` program.
3. A properly commented and formatted version of the `Lab1Part2.java` program.
4. The output from running `Lab1` in the terminal window.
5. The output from running `Lab1Part2` in the terminal window.

In adherence to the Honor Code, students should complete this 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. With the exception of the `Lab1.java` program that you created in the first part of the assignment, deliverables that are nearly identical to the work of others will be taken as evidence of violating the Honor Code.