

**CMPSC 111
Introduction to Computer Science I
Fall 2015**

Practical 5

9 October 2015

**Due in Bitbucket by 14 October 2015 at 11:00 am
“Checkmark” grade**

Summary

As a means to becoming more proficient when editing Java programs, you will customize your `gvim` text editor, observe and document the changes that are evident after customization, and then explore additional commands that you can use in `gvim` to quickly and effectively manipulate Java code. Then, using the “`git add`”, “`git commit`”, and “`git push`” commands you should upload a written reflection on your experiences to your Git repository hosted by Bitbucket.

Review Additional Resources

Since your textbook does not include a detailed discussion about `gvim`, you will need to review some additional print and online resources that explain how to become more adept at editing the text of a Java program. First, please take turns scanning the chapters in the “Practical Vim” book that the course instructor brings to the practical session. If you want to further improve your “essential Vim skills” so that you can “edit text at the speed of thought”, then you should also visit the <http://vimcasts.org/> Web site to watch the screencasts and read the articles about Vim.

Learning About `spf13-vim`

Known as the “ultimate Vim distribution”, `spf13-vim` is a configuration of the `gvim` text editor designed to allow you to make it both highly-customized and easy-to-use. The creator of `spf13-vim`, Steve Francia, explains that this Vim distribution is “designed for programming”, which is exactly how we normally use `gvim` in this class. You can learn more about `spf13-vim` by visiting the Web sites in the below list. As you are reading this material, please ask the course instructor or a teaching assistant if you encounter something that is hard to understand.

- The main `spf13-vim` Web site: <http://vim.spf13.com/>
- Explaining the benefits of `spf13-vim`: <http://spf13.com/post/why-i-use-spf13-vim/>
- The GitHub page for `spf13-vim`: <https://github.com/spf13/spf13-vim>

Installing and Using `spf13-vim`

After you have finished learning more about `spf13-vim` and the features that it provides, you are ready to install it into your home account by typing the following command in your terminal.

```
curl https://j.mp/spf13-vim3 -L > spf13-vim.sh && sh spf13-vim.sh
```

Please make sure that you type the command exactly as it is written; if you do not input this command correctly then you will not be able to improve your configuration of `gvim`. Once you have typed this command you will see that many plugins will be downloaded and installed by a package

manager. Try to observe what is happening and take notes about what you see. Do you recognize the names of any of the `gvim` plugins as they are being installed? What do they do?

Now you are ready to try out your improved version of `gvim`. Using your terminal window, please go into the `practicals/practical01/` directory to find the `Kinetic.java` program that you studied during the first practical assignment. You can edit this file by typing the command “`gvim Kinetic.java`” in your terminal. Please carefully study the new design of `gvim`—what are five ways in which it is now different from the “default” configuration that you were using previously? To best answer this question, you should add some new lines of code to `Kinetic.java`.

There are many steps that you can take to further configure the `gvim` text editor. For instance, by using the “Edit/Color Scheme” menu item you can change the way in which `gvim` uses color to highlight the syntactic elements of a Java program. Additionally, `spf13-vim` installs many plugins that you can use to write Java programs more efficiently. If you study the main `spf13-vim` Web site, you will notice that some of the plugins are activated by pressing the “<Leader>” key when you are in command mode; your current configuration of `gvim` uses the comma key (i.e., “,”) as the leader key. After adjusting the color scheme to suite your taste, you should learn how to use at least one of the plugins that `spf13-vim` has installed. How does your chosen plugin work? What features does it provide? Do you plan to use this plugin on a regular basis? Why or why not?

Learning how to write Java programs in `gvim` is similar to learning a new human language. That is, the `gvim` text editor has its own “language” that you can learn. For instance, pressing the “=” key twice in `gvim`’s command mode will format a line of text with proper indentation. In addition, the “`gg`” and “`G`” commands respectfully move the `gvim` cursor to the top and bottom of your Java program. Knowing these facts, what do you think that the “`gg=G`” command does? Of course, `gvim`’s language includes a wide variety of additional commands like “`u`”, “`d`”, “`dd`”, and “`dip`”. How do these commands allow you to manipulate your Java program?

Completing the Practical Assignment

To finish this assignment and earn a “checkmark”, you should create a `practicals/practical05/` directory in your Bitbucket repository. Then, you should use your newly configured version of `gvim` to create a file called “`responses`”. Inside of this file, you should provide an answer to all of the questions that were posed in this assignment sheet. That is, you can retype the question that you see in the assignment sheet and then furnish your answer below it. For example, one of the questions that the assignment poses is “how does your chosen plugin work?” Finally, you should turn in a screenshot that shows your re-configured `gvim` editing the `Kinetic.java` program.

General Guidelines for Practical Sessions

- **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.
- **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 your writing with other students in the course, text that is nearly identical to, or merely variations on, the work of others will be taken as evidence of violating the Honor Code.