

**CMPSC 112 – Introduction to Computer Science II (Spring 2015)****Prof. John Wenskovitch**<http://cs.allegheny.edu/~jwenskovitch/teaching/CMPSC112>**Lab 1 - Version Control with Git and Bitbucket, and Using Vim as an Integrated Development Environment****Due (via Bitbucket and hard copy) Monday, 26 January 2015****50 points****Lab Goals**

- Connect to Bitbucket
- Learn some Git commands
- Understand the behavior of some Java programs
- Learn the basics of Vim
- Add configuration files and plugins to Vim

**Assignment Details**

Practicing software developers normally use a version control system to manage most of the artifacts produced during the phases of the software development life cycle. In this course, we will always use the Git distributed version control system to manage the files associated with our laboratory assignments. In this laboratory assignment, you will learn how to use the Bitbucket service for managing Git repositories and the `git` command-line tool in the Ubuntu Linux operating system. After connecting to the course's Git repository and creating your own repository, you will compile and run two Java programs, write about their output, and commit your final report to a repository.

Software developers also use an integrated development environment (IDE) to manage various tasks associated with the design, implementation, and testing of data structures and algorithms. In this course, we will use Vim as an IDE. In this laboratory assignment, you will work with your team members to learn about the basic features associated with Vim and then individually prepare your own tutorial that explains how to use basic Vim commands, Vim runtime configuration files, and plugins to support the navigation and manipulation of Java source code.

**Configuring Git and Bitbucket**

During this laboratory assignment and subsequent assignments, we will securely communicate with the `Bitbucket.org` servers that will host all of our projects. In this laboratory assignment, we will perform all of the steps to configure the accounts on the departmental servers and the Bitbucket service. Through this assignment, you should refer to the following Web site for additional information: <https://confluence.atlassian.com/display/BITBUCKET/Bitbucket+101>. As you will

be required to turn in a report describing each step that you finish in this assignment, please be sure to keep a record of all of the steps that you complete and the challenges that you face. You are also responsible for working with a team to ensure that each member of the team is able to successfully complete each of the steps outlined in this assignment.

1. If you have never done so before, you must use the **ssh-keygen** program to create secure-shell keys that you can use to support your communication with the Bitbucket servers. Type **man ssh-keygen** and talk with the members of your team to learn more about how to use this program. What files does **ssh-keygen** produce? Where does this program store those files?
2. If you do not already have a Bitbucket account, please go to the Bitbucket website and create one – make sure that you use your **allegheny.edu** email address so that you can create an unlimited number of free Bitbucket repositories. Then, upload your ssh key to Bitbucket.
3. Now, you need to test to see if you can authenticate with the Bitbucket servers. First, show the course instructor that you have correctly configured your Bitbucket account. Now, ask the instructor to share the course's Git repository with you. Open a terminal window on your workstation and change into the directory where you will store your files for this course. For instance, you might make a **cs112s2015/** directory that will contain the Git repository that I will always use to share files with you. Once you have done so, please type the following command: **git clone git@bitbucket.org:jwenskovitch/cs112s2015-share.git**. If everything worked correctly, you should be able to download all of the files that you will need to use for this laboratory assignment. Please resolve any problems that you encountered by first reviewing the Bitbucket documentation and then discussing the matter with your team. If you are still not able to run the **git clone** command, then please see the instructor.
4. Using your terminal window, you should browse the files that are in this Git repository. In particular, please look in the **labs/lab1/src/** directory and use Vim to study the two Java programs that you find. Remember, the **cd** command allows you to change into a directory.

## Creating a New Repository

Now that you have learned how to clone an existing Git repository, you should make a new repository in the **cs112s2015/** directory that you previously created. First, make a new directory called **cs112s2015-<your user name>**. Then, change into this directory and type the command **git init ..** At this point, you should go into the **cs112s2015-share** repository and use the **cp -r** command to copy the entire **labs/** directory from the **cs112s2015-share** repository to **cs112s2015-<your user name>**. Once the files are in your own Git repository, please use the **git add** and **git commit** commands to add them correctly. If you do not know how to use the **git add** and **git commit** commands in the terminal window, please learn more about them by searching on the Internet, talking about them with your team, and discussing them with the course instructor.

Next, you should use the Bitbucket website to create a repository that has the same name as the local directory and local repository. You must follow Bitbucket's instructions to push the code and tags in your local repository to the remote one. When you are finished with this step, you should see in your browser that the Bitbucket servers are storing the two Java programs. Once

the Git repository contains the correct files, you should share your Bitbucket repository with the course instructor, whose Bitbucket username is `jwenskovitch`.

At this point, you can learn more about Git by consulting websites like <http://try.github.io/> and <http://gitimmersion.com/>. At minimum, you should ensure that you fully understand how to use the following Git commands in the terminal window:

- `git init`
- `git status`
- `git add`
- `git commit`
- `git push`
- `git pull`

## Compiling, Running, and Understanding Java Programs

Once you have mastered the use of Git and version control, you should return to the `labs/lab1/src/` directory that contains the two Java programs. Now, use the Java compiler to compile the `Hooray.java` program. That is, you should type `javac Hooray.java` in the terminal window. Next, you can run this program by typing `java Hooray` in the terminal window. What output does this program produce? Why does it create this output? How do you stop this program?

After compiling, using, and studying the `Hooray` program, you should complete the same steps for the `Weeee.java` program. Go ahead and compile and run this program. What output does it produce? Why does it create this output? How is the output similar to and different from that which was created by the `Hooray` program? Once you have finished studying and understanding these two programs, add comments to the code to explain what they do and how they work. Finally, please make sure that the commented version of each program is correctly committed to your Git version control repository hosted by Bitbucket.

## Using Runtime Configuration Files

It is very easy to configure Vim by writing VimScript in your `.vimrc` and `.gvimrc` files. To complete this phase of the assignment, you should run the `git pull` command inside of your `cs112s2015-share` Git repository. Now, change into the `labs/lab1/src/dotfiles/` directory. What files did you find in this location? Now, you should copy these files from the git repository into the root of your home directory. At this point, you should run the GVim command so that you can see the source code of one of the Java programs that we used earlier in the lab. Do you see that the color scheme is different? If you would like, you can customize the color scheme by using the “Edit” and “Color Scheme” menus. Finally, you should also use GVim to study the VimScript in the `.vimrc` and `.gvimrc` files. Make sure that you and your team members have a basic understanding to these configuration files. What new features does GVim now have?

## Learning the Basics of Vim

Before you start to complete the remainder of this laboratory assignment, you may want to review some of the reasons why people like to use the Vim text editor, as explained at the following website: <http://usevim.com/2012/10/26/why-vim/>. When you are finished learning about some of the reasons behind using Vim, you can return to the GVim window that should still have the source code of a Java program in it. Using your own program and ultimately writing your own tutorial, you should work with your team members to identify, learn, and document some of the basic features that are offered by Vim. For instance, make sure that you know how to perform the following actions:

1. Open, close, and save files in the windows or tabs.
2. Move to the beginning and end of a file.
3. Navigate to specific lines and columns within a file.
4. Enter and exit normal mode.
5. Enter and exit insert and append modes.
6. Select line(s) of text in visual mode.
7. Copy, paste, and delete lines of text.
8. Undo the result of a previous command.
9. Search for and replace specific words in a file.
10. Any additional features that you deem to be useful.

Since we will be using Vim throughout the semester, please make sure that you can easily invoke all of the editor's most important commands. You should take notes and screenshots to demonstrate that you understand how to use basic Vim commands. As you explore how to use Vim, you should prepare content for a tutorial explaining all of the aforementioned tasks.

## Using Plugins to Extend Vim

We will use a variety of Vim plugins to ensure that Vim can operate as a full-fledged integrated development environment when you complete the laboratory assignments and the final project. In this phase of the assignment, you are responsible for learning how to use all of the plugins in the following list. To start learning more about these plugins, load the source code of your `.vimrc` file into GVim and search for a line that starts with the command **Bundle**. For each of the plugins that you are required to investigate, you can visit the associated website, as listed in the `.vimrc`.

Next, you should run the `:BundleInstall` command in GVim. After all of the plugins are correctly installed, your enhanced version of Vim should have many new features! To access these features, you should quit GVim and then run the program again to view the same Java program that you were previously editing. Now, you should prepare a tutorial that explains the inputs, outputs, and behavior of the key features offered by each of the following plugins:

1. Ctrl-P
2. Fugitive
3. MatchIt
4. NERDTree
5. Sneak
6. TComment
7. Tagbar

For instance, when learning how to use the Ctrl-P plugin, you should press the key combination `<ctrl-p>` and then use the interface to navigate the file system and load in new files. Alternatively, you can press `<F11>` and browse the file system and load files with the NERDTree plugin.

## Submission Details

For this assignment, please submit a written description of the procedures you have followed while completing this lab. Also, please upload this written document to your `cs112s2015-<your user name>` repository. Your submission should include the following:

1. A description of the steps that a user must take to configure Git and Bitbucket.
2. A description of the inputs, outputs, and behaviors of the six Git commands listed in the **Creating a New Repository** section.
3. A commented version of the `Hooray.java` and `Weeee.java` programs.
4. A report that clearly responds to the following four prompts:
  - (a) The steps that you took to compile and run both of these programs.
  - (b) The output that each of these programs produce.
  - (c) An explanation for why these programs create the output that they do.
  - (d) A discussion of the similarities and differences between these programs and their outputs.
5. A short tutorial that will explain to a new user the following:
  - (a) A commentary on how Vim uses runtime configuration files.
  - (b) A full-featured description of the basic features associated with the Vim text editor.
  - (c) A complete introduction to the use of the aforementioned Vim plugins.

Before you turn in this assignment, you also must ensure that the course instructor has read access to your BitBucket repository that is named according to the convention `cs112s2015-<your user name>`. Please note that each student in the class is responsible for completing and submitting their own version of this assignment. However, you also will be assigned to work to a team that is tasked with ensuring that all of its members are able to complete each step of the assignment. Team members should make themselves available to each other to answer questions and resolve any problems that develop during the laboratory session. While it is acceptable for members of a team to have high-level conversations, you should not share source code or full command lines with your team members. To ensure that you can communicate effectively, members of each team should sit next to each other in the room. Please see the instructor if you have questions about this policy.