CMPSC 111
Introduction to Computer Science I
Fall 2014

Practical 1
4–5 September 2014
Due: September 10, 2014 by the end of class
"Checkmark" grade

Introduction

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 and practical assignments. In this practical assignment, you will learn how to use the Bitbucket service for managing Git repositories and the `git` command-line tool in the Ubuntu operating system. After connecting to the course's Git repository and creating your own repository, you will compile and run one Java program, capture the program's output, and commit your output to a repository.

Configuring Git and Bitbucket

During this practical assignment and subsequent assignments, we will securely communicate with the Bitbucket.org servers that will host all of our projects. In this practical assignment, we will perform all of the steps to configure the accounts on the departmental servers and the Bitbucket service. Throughout the 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 use Git in the remaining laboratory and practical assignments and during the class sessions, 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 partner to ensure that each of you is able to successfully complete each of the steps outlined in this assignment.

1. If you do not already have a Bitbucket account, please go to the Bitbucket Web site 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 while you are a student.

2. 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. Follow the prompts to create your keys and save them in the default directory (press "Enter" after you are promted: "`Enter file in which to save the key ...   :`", then press "Enter" twice if you do not wish to create a passphrase at this time or type your selected passphrase if you do). Type `man ssh-keygen` and talk with your partner to learn more about this program. What files does `ssh-keygen` produce? Where does this program store these files by default?

   Once both you and your partner have created your ssh keys, you should raise your hand to invite either a teaching assistant or a course instructor to help you with the next steps. First, you must log into Bitbucket and look in the right corner for an account avatar with a down arrow. Click on this blue link and then select the "Manage account" option. Now, scroll down

until you found the "SSH keys" option and upload your ssh key to Bitbucket. You can copy your SSH key by going to the terminal and typing "`cat ~/.ssh/id_rsa.pub`" command.

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 your 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, by opening a terminal window and typing the command "`mkdir cs111F2014`" you could make a `cs111F2014/` directory that will contain the Git repository that the instructor will always use to share files with you. Once you have done so, please type the following command if you are in **sections 1 and 2** of the course:
"`git clone git@bitbucket.org:janyljumadinova/cs111f2014-share.git`",
or the following command if you are in **sections 3 and 4** of the course:
"`git clone git@bitbucket.org:gkapfham/cs111f2014-share.git`".

If everything worked correctly, you should be able to download all of the files that you will need for this practical assignment. Please resolve any problems that you encountered by first reviewing the Bitbucket documentation and then discussing the matter with a teaching assistant. If you are still not able to run `git clone`, then please see a course instructor.

4. Using your terminal window, you should browse the files that are in this Git repository. In particular, please look in the `cs111f2014-share/practicals/practical01` directory and use `gvim` to study the Java program that you find. Remember, the "`cd`" command allows you to change into a directory. So, you could type the following commands to go to the directory that contains today's Java program; make sure you fully understand these steps!

```
cd cs111F2014
cd cs111f2014-share
ls
cd practicals
ls
cd practical01
ls
```

Creating a New Repository

Now that you have learned how to clone an existing Git repository, you should make a new repository in the `cs111F2014/` directory that you previously created. First, make a new directory called `cs111F2014-<your user name>` using the "`mkdir`" command in your terminal window. If you opened a new terminal, then you could type the following commands to create the needed directory; again, make sure that you understand each of these steps, discussing them with your neighbor, a teaching assistant, or one of the course instructors if you are confused.

```
cd cs111F2014
mkdir cs111F2014-<your user name>
cd cs111F2014-<your user name>
```

Once you have changed into this directory you can type the command "`git init .`". Then, you can use the "`mkdir practicals`" command to make a new directory and "`cd practicals`" to change into it. Next, you should again use the "`mkdir`" command to create an additional directory called `practical01`. Please make sure that you completed each of these steps inside of the parent directory called `cs111F2014-<your user name>`; if you are stuck, ask a course instructor for help.

After completing this step you should click on the file icon in the upper left corner to load the Ubuntu file browser. Once the first window displays, you should click the "File/New Window" menu item to load a second graphical file browser. At this point, you should have two file system browsers that are displaying the same content. In the first window, which we will call the "source" window, you should navigate to the `cs111f2014-share` directory by clicking on the correct folder icons. Now, please find the `Kinetic.java` file that is stored inside of the `cs111f2014-share` repository in the "source" window. In the second window, which we will call the "destination" window, you should follow the same process to first navigate to the `cs111F2014-<your user name>` directory, ultimately finding the `practical01` directory. Before you move on to the next step, please have a course instructor or a teaching assistant verify that you have everything set up correctly.

The next step in this practical requires you to use the graphical browser to copy the `Kinetic.java` file from the "source" browser to the "destination" browser, ensuring that the file is transferred from the `cs111f2014-share` repository to the `practical01` directory of the `cs111F2014-<your user name>` repository. Once the files are in your own Git repository, please use the `git add` and `git commit` commands to add them correctly. You can learn how to use the `git add` and `git commit` commands in the terminal window by reviewing the "Git Cheatsheet", discussing them with teaching assistants and course instructors, or searching on the Internet.

Next, you should use the Bitbucket Web site 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 Web browser that the Bitbucket servers are storing the Java program. Once the Git repository contains the correct files, you should share your Bitbucket repository with the course instructor. If you are in sections 1 and 2 your instructor's Bitbucket user name is "janyljumadinova" and if you are in sections 3 and 4 your instructor's Bitbucket user name is "gkapfham".

At this point, you can learn more about Git by consulting Web sites like `http://try.github.io/` and `http://gitimmersion.com/`. After discussion them with a class member and a teaching assistant, you should ensure that you have a basic understanding of the following Git commands:

1. `git init`

2. `git status`

3. `git add`

4. `git commit`

5. `git push`

6. `git pull`

Compiling, Running, and Understanding a Java Program

Once you have mastered the basic use of version control, you should return to the `practicals/practical01` directory—in your version control repository—that contains the Java program. Now, use the Java compiler by typing "`javac Kinetic.java`" in the terminal window. Next, you can run this program by typing "`java Kinetic`" in the terminal window. What output does this program produce? Just like in this week's laboratory assignment, you should practice compiling and running this Java program until you are completely comfortable with these steps.

For this practical assignment, you are not expected to understand all of the details of this program—just do your best to understand lines of the code that look familiar, consulting your text book and talking about it with your classmates and the teaching assistants as needed. If you are up for a challenge, try to locate and fix the defect in this program! If the equation for kinetic energy, denoted K, is $K = \frac{1}{2} \times m \times v^2$, then can you find and fix the defect?

Once you have finished studying this Java program, add comments to the source code to explain what it does and how it works. Again just do your best to explain everything that you understood, while ignoring the parts that did not make sense to you. Most importantly, make sure that a final version of this program is correctly committed to your Git version control repository hosted by Bitbucket; you can do this by going to your repository's directory and correctly using `git add,` `git commit`, and `git push` commands. Additionally, please make sure that you commit the output of the program that you ran in the terminal window. You can accomplish this task by highlighting the program's output, pasting it into a `gvim` window, saving the output in a file with an appropriate name, and using the correct `git add`, `git commit`, and `git push` commands to store it on Bitbucket. If you are not able to complete this step, then please seek assistance from a teaching assistant or a course instructor so that you can finish by the assignment's stated deadline.