CMPSC 111
Introduction to Computer Science I
Spring 2017

Practical 1
Assigned: Friday, January 20, 2017
Due: Friday, January 27, 2017 at the start 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.

Please carefully adhere to the following guidelines for success when you are completing this practical assignment and all of the subsequent laboratory and practical assignments.

- **If possible, use the laboratory computers.** If it is absolutely necessary for you to work on a different machine, be sure to regularly transfer your programs to the Alden machines and check their correctness. Please remember that, as stated in the syllabus, students should try to complete assignments using the specialized workstations in the laboratory. If you cannot use a laboratory computer, then please carefully explain the setup of your laptop to a teaching assistant or the course instructor when you are asking questions.
- **Follow each step carefully.** Slowly read each sentence in every assignment sheet, making sure that you precisely follow each instruction. Take notes about each step that you attempt, recording your questions and ideas and the challenges that you faced. If you are stuck, then please tell a teaching assistant or instructor what step you recently completed.
- **Regularly ask and answer questions.** Please log into Slack at the start of a laboratory or practical session and then join the appropriate channel. If you have a question about one of the steps in an assignment, then you can post it to the designated channel. Or, you can ask a student sitting next to you or talk with a teaching assistant or the course instructor.
- **Store your files in Git**. Starting with this laboratory assignment, you will be responsible for storing all of your files in a Git repository. Please verify that you have saved your source code in your Git repository by typing “`git status`” and ensuring everything is up to date.
- **Keep all of your files!** Don’t delete your programs, output files, 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 your 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.

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/bitbucket/tutorial-learn-git-with-bitbucket-cloud-759857287.html`. 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 communicating 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, then 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 prompted: "`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 you and your partner have created your ssh keys, you should raise your hand to invite either a teaching assistant or the 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 "Bitbucket Settings" 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 check to see if you can "authenticate" yourself 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 computer and change into the directory where you will store your files for this course. If needed, you can open a terminal window and type the command "`mkdir cs111S2017`" to make a `cs111S2017/` directory that will contain the Git repository that the instructor will always use to share files with you. Note that if you correctly followed the instructions from the previous laboratory assignment then you will have already created this directory! Once you have finished this step, please type the following command in your terminal "`git clone git@bitbucket.org:gkapfham/cs111s2017-share.git`".

   If everything worked correctly, you should be able to download the single Java file 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 the course instructor.

4. Again using your terminal, you should browse the files that are in this new Git repository. In particular, please look in the `cs111s2017-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 these commands to go to the directory that contains today's Java program; make sure that you fully understand these steps!

```
cd cs111S2017
cd cs111s2017-share
ls
cd practicals
ls
cd practical01
ls
```

Congratulations on completing these steps to access the `cs111S2017-share` repository that we will use throughout the semester! Please make sure that, before you finish this practical assignment, you are sure that you are correctly connected to this version control repository. Students who cannot easily access this repository will risk falling behind during the semester as the course instructor will use this means of code delivery during all of the remaining class, laboratory, and practical sessions. If you are struggling with this step or the next one, then don't get frustrated—these tasks are difficult and thus often requiring repeated attempts on your part. Ask for help if you need it!

Creating a New Repository

To create your own Git repository that you will use to share files with the course instructor, you will need to return to you web browser and log into Bitbucket if you have not done so already. Now, click the "Repositories" link at the top of your browser window and then click "Create repository". Give your repository the name "`cs111S2017-<your user name>`" where "`<your user name>`" is written to remind you to type your own user name and the quotation marks in this paragraph are there to designate that these are words that you look for or input directly to the browser.

Now, the system will allow you to create a `README` file. Deleting everything except the first line that is inside of the text area, edit the description of your repository by adding in fun details about yourself and your learning goals for this course. Once you have finished editing this file, you should save it and then enter a descriptive message about the changes that you have just made and, finally, click the "Commit" button. Now, find and click the "three dot icon" and then click the "Clone" button. Next, you should copy this entire command and, making sure that you are first in the `cs112S2017` directory, paste the command and then run it. Now, you should see that the new files for your repository are on your computer. Next, change into "`cs111S2017-<your user name>`" and use `mkdir` to create a `practicals` directory that contains a `practical01` directory.

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 `cs111s2017-share` directory by clicking on the correct folder icons. Now, please find the `Kinetic.java` file that is stored inside of the `cs111s2017-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 cs111S2017-<your user name> directory, ultimately finding the practical01/ directory. Before you move on to the next step, please have the 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" window to the "destination" window, ensuring that the file is transferred from the cs111s2017-share repository to the practical01/ directory of the cs111S2017-<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 instructor, or searching on the Internet.

Now, you can learn more about Git by consulting web sites like http://try.github.io/. After discussing them with the instructor and a teaching assistant, you should ensure that you have a basic understanding of these Git commands: git status, git add, git commit, git push, and git pull. Finally, use Bitbucket in the browser to share your Git repository with the instructor.

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 use your intuition to best understand lines of the code that look familiar, consulting the instructor 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.

Since this is your first practical assignment and you are still learning how to use the appropriate hardware and software, don't become frustrated if you make a mistake. Instead, use your mistakes as an opportunity for learning both about the necessary technology and the background and expertise of the other students in the class, the teaching assistants, and the course instructor. Remember, you can use Slack to talk with the instructor by using "@gkapfham" in a channel.