CST8116 Assignment 04

Using Git within Eclipse.

# Instructions

* This assignment focuses on using the Git version control system, within an Eclipse Environment.
* **Expectation**: you have already reviewed and completed instructions for Git installation and configuration
  + Installing and initial configuration: See Hybrid 05 content area, within week 05 lecture notes content area
  + Using Git in Eclipse: See Hybrid 08 content area, within week 10 lecture notes content area
    - The handout here will walk you through this lab assignment as well.
  + **If you have not installed Git, or worked through the handouts for Hybrid 08 do so now.**
* This assignment handout has brief step-by-step instructions
* You will need to screen shot your running program
* You will need to screen shot your Git History for the project in Eclipse
* Submit your source code file, and your Git Repository as a zip archive.

(Follow Lab Professors Directions)

# Steps

## Create an Eclipse Project

* Name the project: Assignment 4

## Create a Git Repository for the project

* Create a Git Repository using Eclipse for your Assignment 4 Project
* Create the following location on your hard drive for the repository:

**C:\CST8116\_Assignment\_04\_Your\_Name**

Your\_Name should be your full name, as see in ACSIS separating each name with underscores.

* You will need to zip this folder and submit it after completing all of the steps in this lab.

## Add the Project files to the Git Staging area

* Add your project to the Git staging area within Eclipse (See hybrid handout(s) )

## Verify .ignore file

* Make sure that the .ignore file has an entry for /bin/
* If you cannot see the .ignore file reference the hybrid handout on how to modify the package explorer filters.

## First Commit

* Commit the initial project files
  + Enter a commit message   
    “New Project Assignment04 by *YourFullName*”
  + Omit the double-quotes.
  + replace YourFullName with your actual name
  + Add a signed-off by entry with your name and email using the interface (see hybrid handout)
  + Commit.

## Create a source code file and add to version control

* Add a class to the project naming it Assignment04*YourFullName* where *YourFullName* is your full name as it appears in ACSIS. For Example Stanley Pieda would name the class File Assignment04StanleyPieda
* Check off the box to create a main method.
* Right-click on the file and add it to the Git Staging Area

## Second Commit

* Commit the new source code file.
  + Enter a commit message  
    “Added source code file Assignment04*YourFullName* with method main”
  + Omit the double-quotes.
  + replace YourFullName with your actual name
  + Add a signed-off by entry with your name and email using the interface (see hybrid handout)
  + Commit.

## Copy starter code into source code file

* Copy the starter code located here (also included in a text file) into method main.
* Modify the code so that it prints out your name, instead of Junior Programmer
* Add programmer comments to the file in the usual manner for our course.
* Note:
  + You will be asked to fix a mistake in the loop condition in a later step.
  + Currently if you run the program as presented here, it will crash (this is on purpose).

## int[] primes = { 2, 3, 5, 7 };

## for(int index = 0; index <= primes.length; index++) {

## System.out.println(primes[index]);

## }

## System.out.println("Program by Junior Programmer");

## Save the changes to the file. (Just normal Save button in Eclipse).

## Third Commit

* Commit the changed source code file.
  + Enter a commit message  
    “Added programmer comments, and starter code to method main.”
  + Omit the double-quotes.
  + Add a signed-off by entry with your name and email using the interface (see hybrid handout)
  + Commit.

## Run the program / fix the bug / run the program & Screen Shot.

* Run the program and observe the error message.
* Fix the problem with the loop condition.
* Run the program and take a screen shot for your MS Word document.

## Fourth Commit

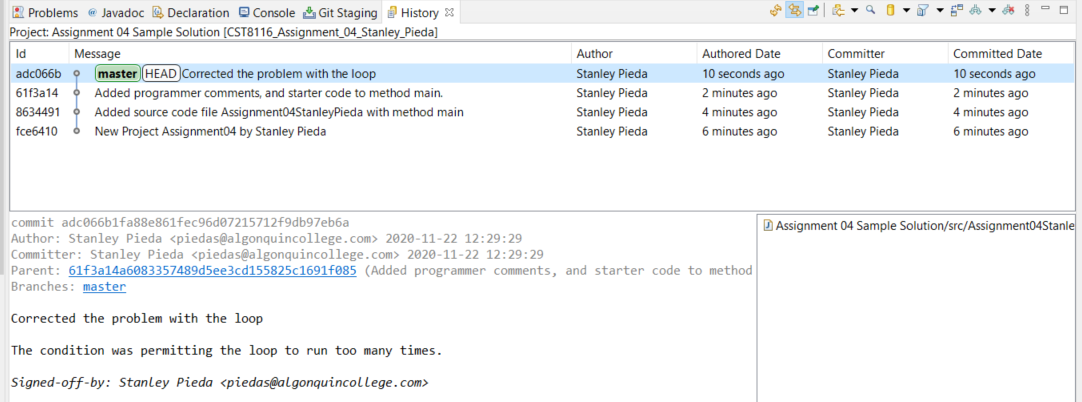
* Commit the changed source code file.
  + Enter commit messages, use Separate lines, blank line between each.   
    “Corrected the problem with the loop.”

“The condition was permitting the loop to run too many times.”

* + Omit the double-quotes.
  + Add a signed-off by entry with your name and email using the interface (see hybrid handout)
  + Commit.

## Take screen shot of Git history in Eclipse

* View the Git History in Eclipse.
* Ensure that your Project (not just the source code file) is selected in the Package Explorer so that the full history is shown.
* Take a screen shot that shows your commit: id, message, author, authored date, committer, and committed date.
* This is an example created by Stanley Pieda



Submit MS Word document with screen shots, your Java file, and a zip archive of the Git Repository you created.

Follow any additional instructions provided by your lab professor.

Your lab professor may verify your repository using command line tools, e.g. >git log to determine that it closely matches your screen shot taken from Eclipse.

# Microsoft Word Document Format

See the template example, suggested headings below

Screen Shot Running Program

Screen Shot Git History in Eclipse

# Grading (10 points)

|  |  |  |  |
| --- | --- | --- | --- |
| Criteria | Missing / Poor (0) | Below Expectations (1) | Meets Expectations (2) |
| Screen Shot, Program Execution | Missing or poorly done (missing student name in screen shot) | Shows program executing successfully, may only have part of student’s name in output. | Shows program executing successfully, student’s full name as part of output. |
| Screen Shot, Git History | Missing or poorly done (missing student name in screen shot) | Shows some but not all of the requested commits, or commit text not correct. May only have part of the student’s name in the screen shot. | Shows all four requested commits and closely matches sample screen shot. Student’s full name is visible in the screen shot. |
| Source Code Comments | Missing or poorly done | Missing some of: Programmer comments at top of file, descriptive comment for class and method name. | All completed: Programmer comments at top of file, descriptive comment for class and method name. |
| Source Code Fixed | Missing or poorly done or is un-modified starter code. | Problem with loop condition not fixed correctly; starter code was copied into method main. | Problem with loop condition is fixed; starter code was copied into method main. |
| Git Repository | Missing from submission or only a project folder and not a full Git Repository. | Does not have requested folder name for repository, git log command may show inconsistencies when compared to screen shot. | Has requested folder name for repository, git log command closely matches the history seen in the screen shot. |

# Submission Requirements

* Upload these items to Brightspace by the due date:
* MS Word document containing screen shots.
* Java file with programmer comments and source code.
* Assignment 04 Git Repository within a zip archive (CST8116\_Assignment\_04\_*Your\_Name*.zip)
  + *Your\_Name* should be your full name as seen in ACSIS.
* Follow your lab professor’s instructions regarding lab submissions for their lab section.

# Appendix: Sample **git log**

* Students are not required to submit a screen shot like this one, it is here for demonstration only.
* Use the space-bar if there are multiple pages of output.
* Enter a ‘q’ for quit, to exit the log tool.

# 