FA23: CMPE-131 Sec 01 - Software Engr I HW 2: Git

Nutthawat Panyangnoi

TOTAL POINTS

15 / 15

QUESTION 1

1 L01 **1 / 1**

✓ **+ 1 pts** *Correct*

+ 0 pts Needs Revision

QUESTION 2

2 L01 **1 / 1**

✓ **+ 1 pts** *Correct*

+ 0 pts Needs revision

QUESTION 3

3 L01 **1 / 1**

✓ **+ 1 pts** *Correct*

+ 0 pts need revisions

QUESTION 4

4 L01 **1 / 1**

✓ **+ 1 pts** *Correct*

+ 0 pts Needs Revisions

QUESTION 5

5 L01 **1 / 1**

✓ **+ 1 pts** *Correct*

+ 0 pts Need Revisions

QUESTION 6

6 L02 **1 / 1**

✓ **+ 1 pts** *Correct*

+ 0 pts Needs Revisions

QUESTION 7

7 L02 **1 / 1**

✓ **+ 1 pts** *Correct*

+ 0 pts Needs revision

QUESTION 8

8 L02 **1 / 1**

✓ **+ 1 pts** *Correct*

+ 0 pts Needs revision

QUESTION 9

9 L03 **1 / 1**

✓ **+ 1 pts** *Correct*

+ 0 pts Needs revision

QUESTION 10

10 L03 **1 / 1**

✓ **+ 1 pts** *Correct*

+ 0 pts Needs Revision

QUESTION 11

11 L03 **1 / 1**

✓ **+ 1 pts** *Correct*

+ 0 pts Needs revisions

QUESTION 12

12 L04 **1 / 1**

✓ **+ 1 pts** *Correct*

+ 0 pts Needs revisions

QUESTION 13

13 L04 **1 / 1**

✓ **+ 1 pts** *Correct*

+ 0 pts Needs Revision

QUESTION 14

14 L04 **1 / 1**

✓ **+ 1 pts** *Correct*

+ 0 pts Needs revisions

QUESTION 15

15 L04 **1 / 1**

✓ **+ 1 pts** *Correct*

+ 0 pts Needs Revisions

HW 2: Git

CMPE 131-S01: Software Engineering I
San Jose Staté University
Due Tuesday, September 12, 2023

Full Name **Nutthawat Panyangnoi**

Student ID **016381040**

Introduction

Objective: How to work with git in the Linux command line.

Learning Outcomes

1. Understand what are the basic git commands (pass 5/5)
2. Understand and apply commands to navigate/create/delete git branches(pass 3/3)
3. Know how to access git repositories locally and remotely (pass 3/3)
4. Analyze and write advance git commands (pass 3/4)

Grade

- HP: Pass 4 learning outcomes
- LP: Pass 2 learning outcomes
- NR: Pass 1 or less learning outcomes

Help

If you're in doubt ask in #help on discord or create a ticket in #terminal with `/ticket open 'topic'`. It's only too late to ask after the deadline.

Note

When we ask for Hash only use the first 4 characters.

LO1: Understand what are the basic linux commands. Pass 5/5 questions.

1. (L01) Create a repository and commit files.

Instructions: Follow the instructions below in a Linux terminal and **write down all of the commands** required to accomplish the task.

1. Create a folder named `CMPE131S01`
2. Move into `CMPE131S01`
3. Add a file named `hi.txt` with the text 'Hello Class!'
4. Initialize the git repository and commit the file `hi.txt`

1. Create a floder with command **mkdir CMPE131S01**

2. I think you mean go inside floder **cd CMPE131S01**

3. Add a file name hi.txt and text 'Hello Class' **echo "Hello Class!" > hi.txt**

4 intitialize git and commit

git init
git add hi.txt
git commit -m "first commit"

2. (L01) Add a new file to git repository.

Instructions: Follow the instructions below in a Linux terminal and answer the question at the end. This question follows from Q1.

1. Create a branch at the current commit called `new-feature`.
2. Now, move to the `new-feature` branch. Note: step 1 and step 2 should be accomplished as *two* steps.
3. Commit 'Hi Class' in `hi.txt` to the `new-feature` branch

Write all of the commands used for steps 1-3. Next, write the command you would use to merge `main` and `new-feature`. You want main to contain the changes from the `new-feature` branch.

1. git branch new-feature or git checkout -b new-feature

2. git checkout new-feature

3. echo "Hi Class" >>hi.txt
then, git add hi.txt
Next, git commit -m 'edit text file to Hi class'

Next move to main **git checkout main**

Now it merge to main **git merge new-feature**

make a commit show update **git commit -am "Merge new-feature branch to main "**

3. (L01) Git history.

Instructions: Follow the instructions below in a Linux terminal and **write down all of the commands** required to accomplish the task.

1. Write the git command to view the git repository history.
2. Write the command to view the history but where the commit message is shown as *one* line.
3. Write the git command to view the git repository history, but you also see the branching structure. Reminder this should work in the Linux terminal.

1.Git log

2.Git log - -oneline

3. Git log - -graph

4. (L01) What is the git command to unstage all of the files that are currently staged.

This git command will unstage all of files **git reset**

5. (L01) What is the git command you would use to find out if you have any new changes that need to be committed in your git repository?

This git command will check all the status in git of your repository **git status**

LO2: Understand and apply commands to navigate/create/delete git branches(pass 3/3)

6. (L02) Changing branches.

Instructions: Follow the instructions below in a Linux terminal and **write out all of the commands** required to follow the instructions (step 1-3).

[This question follows from **Q2**]

1. Create a git branch at the first commit of your git repository and call it `debug-44`.
2. Write the command to switch you to `debug-44`.
3. Write the command to move you back to the `main` branch.

Note my local hash a46a6aded394a628d48da55977ee04e282e26f09

1. git branch debug-44 a46a6

2. git checkout debug-44

3. git checkout main

7. (L02) Delete branches.

What is the command to delete the `debug-44` branch?

git branch -d debug-44

8. (L02) View branches.

Assuming you connected your git repository to github, write the git command to view all of the branches on your local machine **and** on Github.

This command will allow user to see all local branch and remote branch **git branch -a**

LO3: Know how to access git repositories locally and remotely (pass 3/3)

9. (L03) Push changes.

Instructions: Follow the instructions below in a Linux terminal and **write out all of the commands** required to follow steps 1-4

[This question follows from **Q2**]

1. Create a public Github repository.
2. Follow the Github instructions for an *existing* github repository.
3. Push changes from local `main` branch to Github `main` branch (remote).
4. Push changes from local `new-feature` branch to github `new-feature` branch (remote).

1. mkdir myRepository $ cd myRepository $git init $ git add <file> $ git commit -m "first commit"

2. $ git remote add origin <Link URL github> $ git push -u origin master

3. $ git branch $ git checkout main $ git add <file> $ git commit -m " modified .." $ git push -u origin main

4. $ git checkout -b new-feature $ git checkout new-feature

$ git add <file> $ git commit -m " make a change"

$ git push -u origin new-feature

10. (L03) Pull changes.

Assume your teammate made some commits into the `new-feature` branch. What is the git command to pull the changes from the `new-feature` branch on Github onto your local `new-feature` branch?

First, switch to new-feature branch locally **git checkout new-feature**

Then, pull the changes from the github (new-feature branch) to your local git new-feature branch **git pull origin new-feature**

View all the changes that you teammate contributed **git log**

11. (L03) Cloning.

What is the `git clone` command used for? Write in your own words.

git clone is a command that used for create a copy downloadable file from an specific existing repository.

With little modified and specific this command can work on local git and also on github repository

LO4: Analyze and write advance git commands (pass 3/4)

12. (L04) Explore a large git repository (Pt. 1)

Instructions: Follow the instructions below in a Linux terminal and answer the question at the end. Note: some of the commands may take a several minutes (depending on your internet connection).

1. git clone https://github.com/torvalds/linux
2. cd linux

Answer the following questions:

- What is the date and the name of the author of commit with hash 7d92e89363755978c616c8d9d9f8961989e62be8?
  - Write answer here using the date format MM/DD/YY.

To get a data **git show 7d92e89363755978c616c8d9d9f8961989e62be8**

Author that commit with this hash is Author: Zhangjin Wu <falcon@tinylab.org> Date: 07/06/23

13. (L04) Explore a large git repository (Pt. 2)

What is the oldest commit of the Linux repository? Write hash, author name, and date.

Using command to get reverse order find the oldest day of commit **git log - -reverse**

Oldest hash **1da177e4c3f41524e886b7f1b8a0c1fc7321cac2**

By Author: Linus Torvalds <torvalds@ppc970.osdl.org>
Date: Sat Apr 16 15:20:36 2005 -0700

Date format MM/DD/YY **04/16/05**

14. (L04) Merge conflict

Given the git repository at Q2. How would you create a merge conflict between `main` branch and `new-feature`? Write out all of the commands.

1. Select to main branch	**git checkout main**
2. Make a change on file	**vim hi.txt**
3. Edit and replace word "Hello Class!" to "Hello everyone!"	**echo "Hello everyone! " > hi.txt or nano hi.txt and edit**
4. git add	**git add hi.txt**
5. commit	**git commit - m 'hi.txt has been edited'**
6. switch to new branch	**git checkout new-feature**
7. Edit and replace word "Hello Class!" to "Hello"	**echo "Hello " > hi.txt or vim hi.txt and edit**
8. git add	**git add hi.txt**
9. commit to branch	**git commit - m 'hi.txt has been edited in new-feature branch'**
10 .Merge	**git merge new-feature**

This will result conflict between main branch and new-feature

15. (L04) Fixing conflicts

Write all of the methods you could use to fix a merge conflict, such as the one you created in **Q14**.

To fixing a conflict select each branch and look for conflict file **In this case is hi.txt in main and hi.txt in new-feature**

1 .Check git status first to indentify the conflict file **git status**

2. You can now decide which heading to keep or edit the file to combine the changes.
3. In either case, remove the markers that indicate the beginning and end of the changes, leaving only the content u want in the file

4. After resolve conflict re-add the file **git add hi.txt**

5.Commit new change **git commit -m 'fix merge conflict file'**