**LEARN GIT**

**Poem Fiasco**

In this project, changes have been made to a series of poems and you want to change them back.

Poems Fiasco

1.

The file **road-not-taken.txt** does not look right at all! Perhaps a rival poet snuck in and changed it while you were getting coffee. Good thing you have been committing often.

Checkout the HEAD version of **road-not-taken.txt** to discard changes made to the working directory.

Close and re-open the file to see the result.

**Ans:** Git checkout HEAD road-not-taken.txt

2.

Now, finish the poem by adding a line. Here is a suggestion:

And that has made all the difference. Then click Save.

**Ans:** add some lines and save the txt file.

3.

Look at **oven-bird.txt** to see if it has also been tampered with.

Indeed, it has! We will want to discard changes in the working directory again.

There is a commonly used shortcut for this command:

git checkout -- filename

It does the same exact thing that git checkout HEAD filename does.

**Ans:** Use git checkout -- oven-bird.txt or git checkout HEAD oven-bird.txt can be used to discard any changes in the working directory.

4.

Now, finish “Oven Bird” by adding a line. Is what to make of a diminished thing. Then click Save.

**Ans:** Add some lines and save.

5.

Click on **fire-and-ice.txt.**

This file has not been altered, but just to be sure, check the diff for this file. Then, add these last two lines to the poem: Is also okay. And would suffice.

Ans: git diff

will give the difference of file can be seen in both working directory and staging area

6.

Now that you have restored and completed road-not-taken.txt and oven-bird.txt and added a line to fire-and-ice.txt, add all three of the files to the staging area with a single command.

Ans: git add filename1 filename2 filename3 (or)

git add . does the same thing

This will add all the files from working directory to staging area

7.

fire-and-ice.txt could be better. Before you commit, remove this file from the staging area.

**Ans:** git reset HEAD filename

Will bring back from staging area to working directory

8.

Now that you have removed fire-and-ice.txt, make a commit.

**Ans:** git commit -m” Fire poem is un-staged and poem road and oven modified”

9.

You get the crazy idea to change your poems in a big way.

Make some drastic changes to each of the three poems. Remember to click Save after each file change.

**Ans:** make some modifications and save it

10.

Now add all three files to the staging area.

**Ans:** git add.

11.

Make a commit.

**Ans:** git commit -m “All poems modified and staged, ready to commit”

12.

A little later you look at the current state of your poems and regret your last commit.

Reset your Git project to the commit before you made those drastic changes.

**Ans:** git reset 4106ea

The syntax is <git reset last commit\_SHA>

13.

There is a problem: you reset HEAD to a previous commit, but the changes you want to get rid of are still in the working directory.

What Git backtracking command that you already know can discard changes to the working directory, restoring the files to the way they look in the HEAD commit?

**Ans:** git checkout HEAD filename1 filename2 filename3 --- will bring back from staging area to working directory.