Assignment 1: Initialize a new Git repository in a directory of your choice. Add a simple text file to the repository and make the first commit.

# # Git Initialization

Git Init

# #Adding a text file

# using touch command

touch gitFile.txt

* Or using cat command

cat >gitFile2.txt

# #Steps to commit files

* Firstly, we have to add a file into staging area:
  + This can be done with either using only current file name needed to commit:

git add gitFile.txt

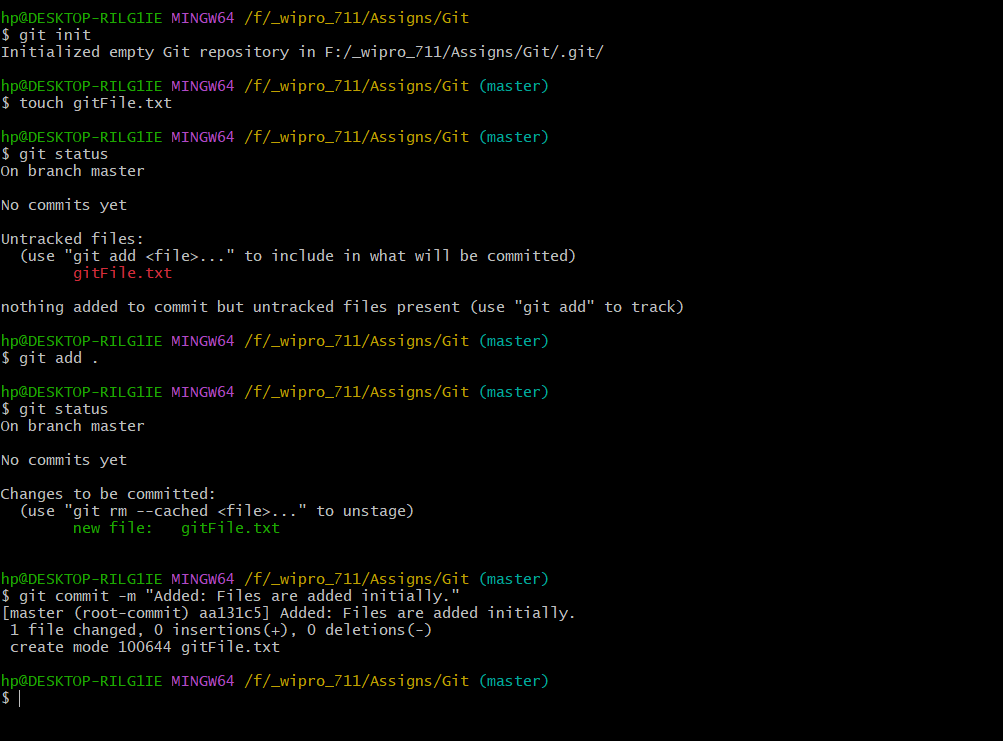
* + Or we can add all files of directory to staging area:

git add .

* After adding to stage area we can commit our changes using:

git commit -m “Added: files into git initially”

# # Git Bash Process as follows:



Assignment 2: Branch Creation and Switching

Create a new branch named 'feature' and switch to it. Make changes in the 'feature' branch and commit them.

# Create a New Branch and Switching on it:

* Using **checkout** only to switch and create branch at same time:

git checkout -b feature

* Using checkout and branch only to switch and create branch at same time:
  + **branch** command to create branch named feature

Git branch feature

* + checkout command to switch to feature branch

git checkout feature

* + **Alternatively, switch** command to switch to feature

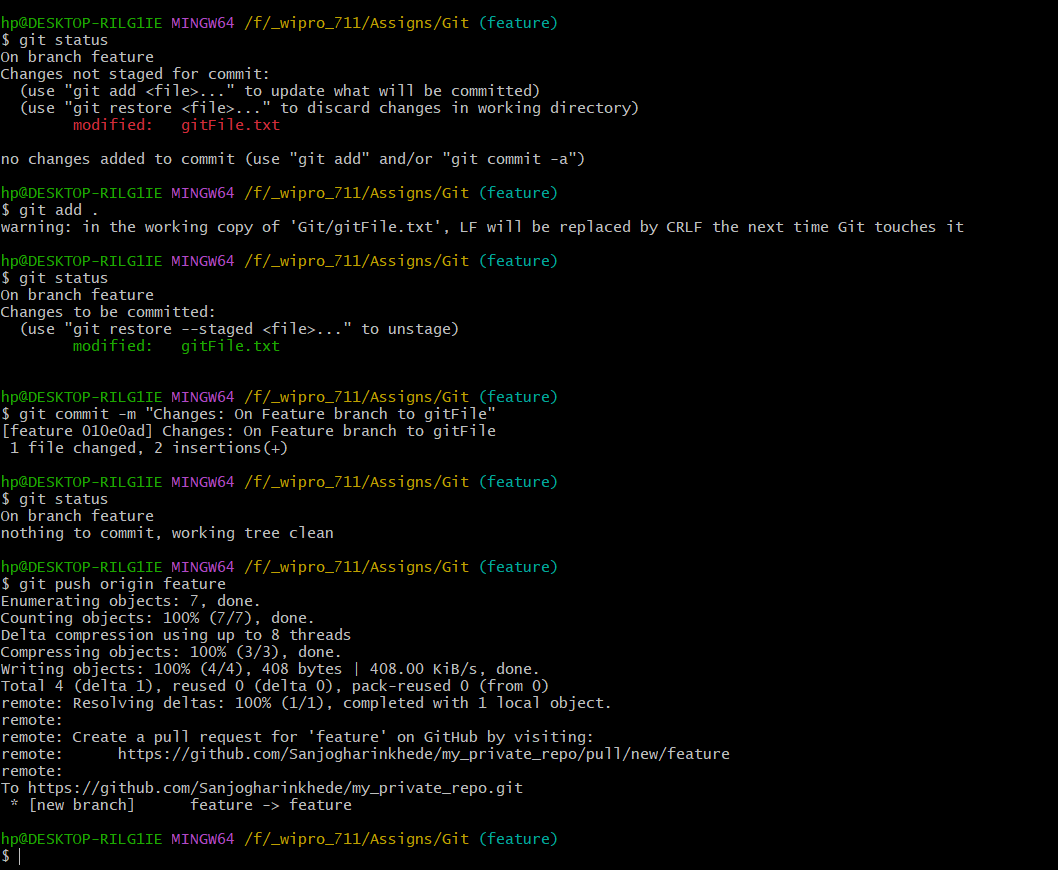
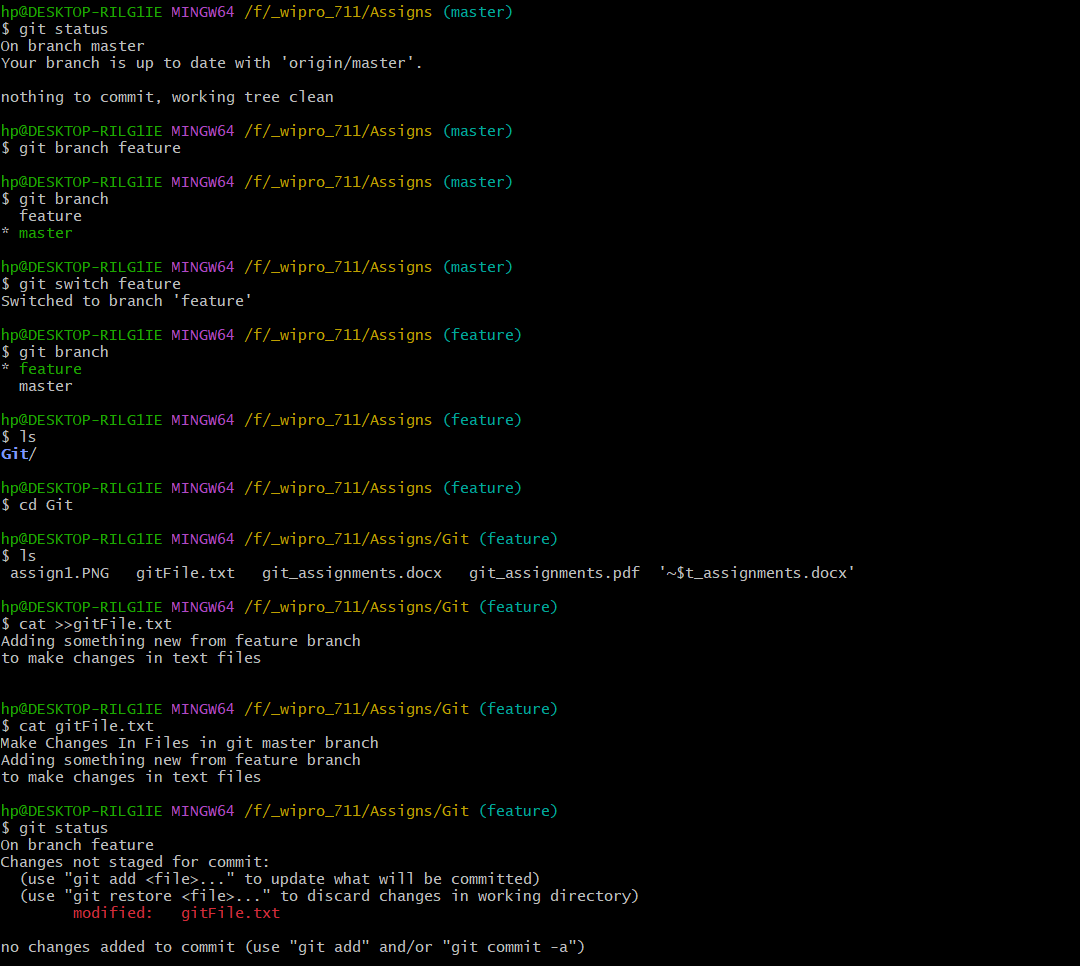
git switch feature

* After switch and changing files, we use add and commit to save changes:

git add .

git commit -m “Changes: Files are Changed on feature branch”

# # Git Bash Process as follows:



Assignment 3: Feature Branches and Hotfixes

Create a 'hotfix' branch to fix an issue in the main code. Merge the 'hotfix' branch into 'main' ensuring that the issue is resolved.

# Fixing a bug on hotfix and merging to master(main):

* Hotfixes are usually created from master branch and switch to hotfix with

latest version of it.

git checkout master

git pull origin master

git checkout -b hotfix

* Make necessary changes on code and save changes locally then commit:

git add .

git commit -m "Hotfix: Bug on code with msg"

* After Fixing issues and testing, merge branch to main for that we have to checkout to main and there use merge request or if having permissions merge hotfix branch into master

git checkout main

git merge hotfix

* Usually in production we make tags for these hotfixes (say version 1.1.0), can be done using and push to master branch with tags

git tag -a v1.1.0 -m "Release: Fix critical issue"

git push origin main --tags

* Usually after fixes we delete the hotfix branch:

git branch -d hotfix

git push origin --delete hotfix

# # Git Bash Process as follows:

