**Complete Git and GitHub Masterclass : Beginner to Git Expert**

**WHOLE PRACTICE ASSIGNMENT**: Initialize a new repository for the project using Git. (1HR 30MINS)

1. Use git add command to stage a new file. Use “git commit” command with commit message to commit changes in the git repository. Check the status using “git status” command. Check commits history in Git Repository using git log.
2. Create an account in GitHub. Use git pull command to pull changes from GitHub and git push command to push local changes into GitHub repository.
3. Create a feature branch in the git repository using git checkout –b command and modify the file. Commit the changes. Switch to the master branch. Use git diff command to view differences between master branch and feature branch. Use git difftool command to view differences in the p4merge tool.
4. Use p4merge tool to resolve the merge conflicts. Create Lightweight and Annotated tags for the commits. Create a branch from the stash using git stash branch command to move the changes to new branch.
5. git init to initialize local repository.
6. git add .

to add all changes in present directory to stage.

1. git status

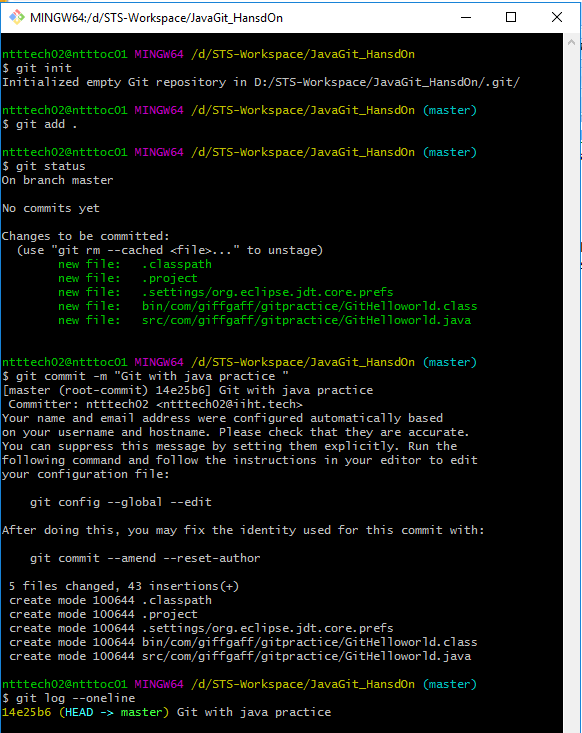
to know changes not pushed from working directory to staging.

1. git commit –m “Git with Java practice”

to commit changes, which creates a commit Id and ready to be pushed to repository.

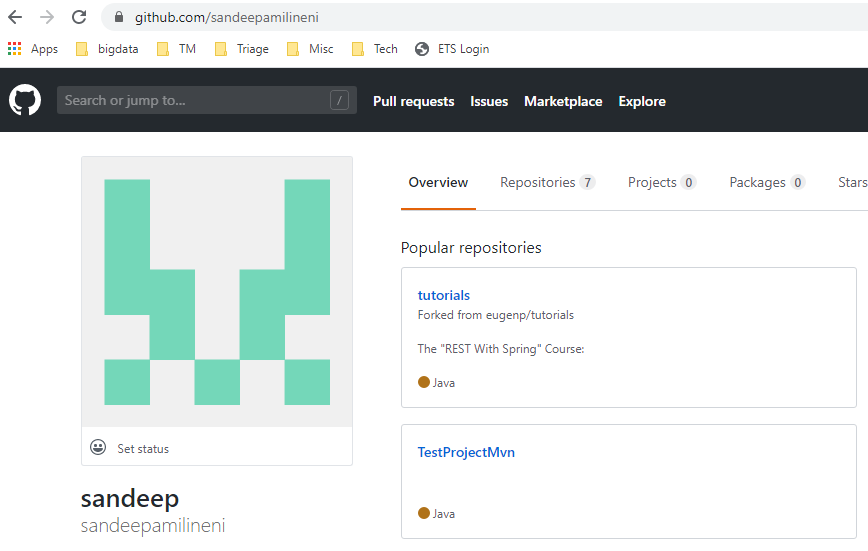
1. git log

to know the list of commits done on remote and local repositories and are not undone

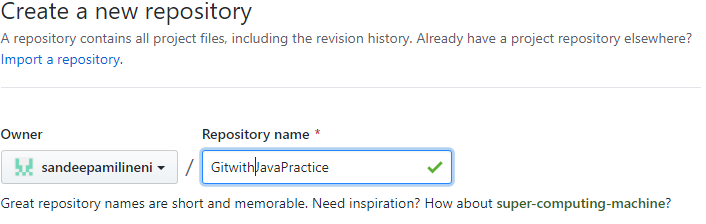


b.

**1. Create an account in GitHub.**

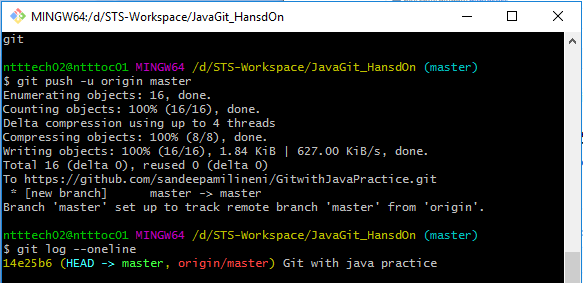


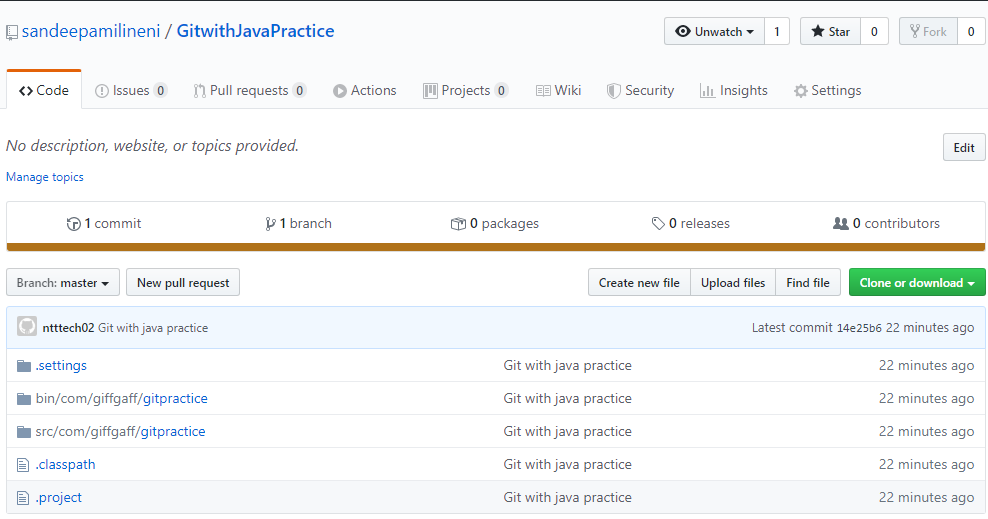
**2. Create a repository on GitHub**



**3. Push changes to remote repository on GitHub from local local repository.**

**g**it push –u origin master

****

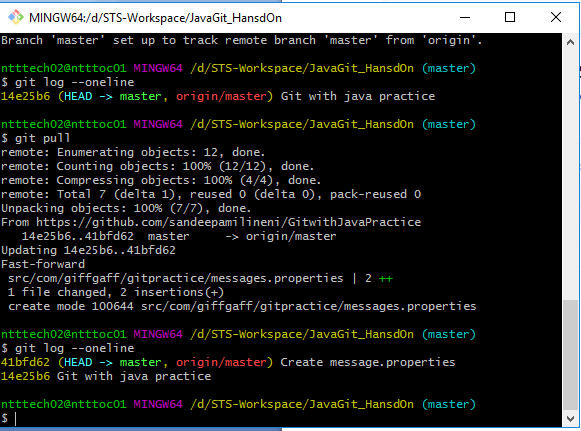


# 4.Make changes in remote repository [sandeepamilineni](https://github.com/sandeepamilineni)/[GitwithJavaPractice](https://github.com/sandeepamilineni/GitwithJavaPractice)

# Created file messages.properties on the remote repository to demo on git pull

# 

Git pull



**Create a feature branch JavaSubBranch and make changes to files and commit**

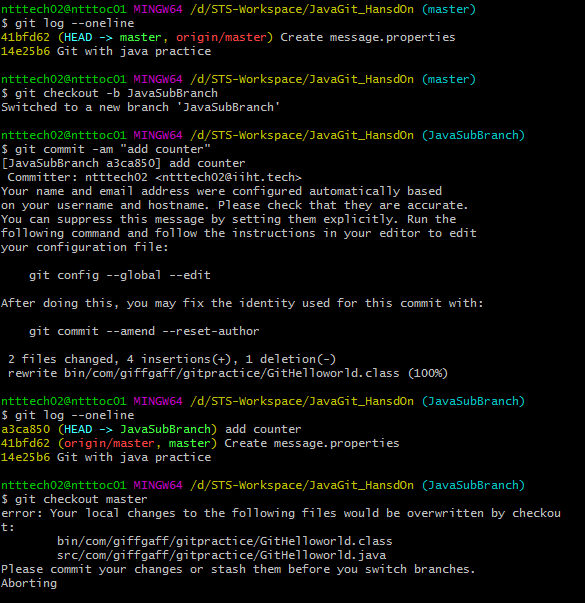
git checkout master

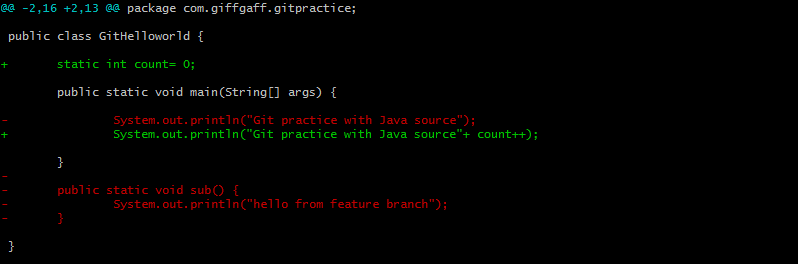
To see the differences between Master and Feature branch

**git diff JavaSubBranch** (from Master branch)

(or)

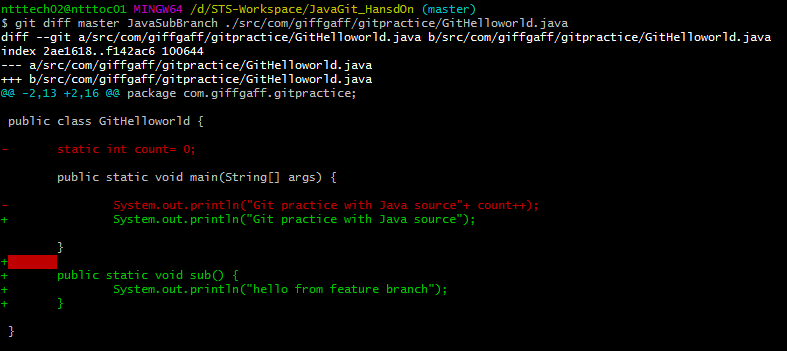
**git diff master..JavaSubBranch**





To see the difference at file level

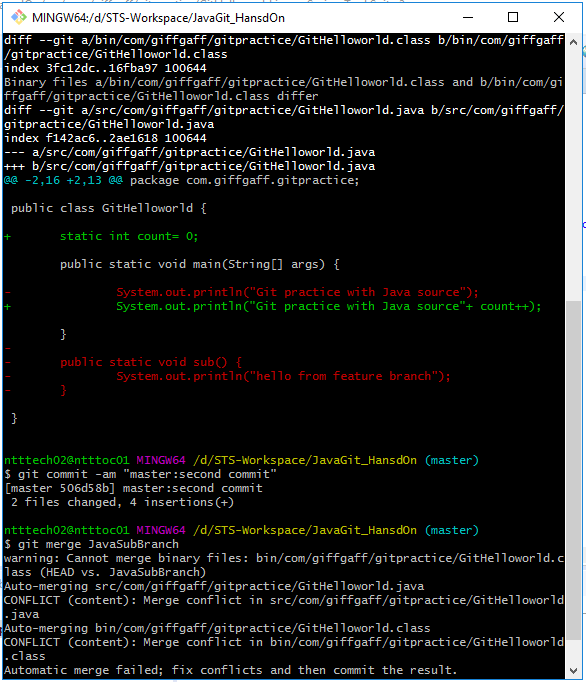
git diff master JavaSubBranch .**/src/com/giffgaff/gitpractice/GitHelloworld.java**



**Switch to Master branch and make changes to Master branch files and commit.**

git merge JavaSubBranch

Trying to merging feature branch with Master shows message as the conflicts cannot be resolved automatically.



**Install p4merge tool** and configure it as git diff and merge assistance tool using following configuration either from git command line or .gitconfig file in user directory

**Git command line**

git config --global diff.tool p4merge

git config --global difftool.p4merge.path 'C:\Program Files\Perforce\p4merge.exe'

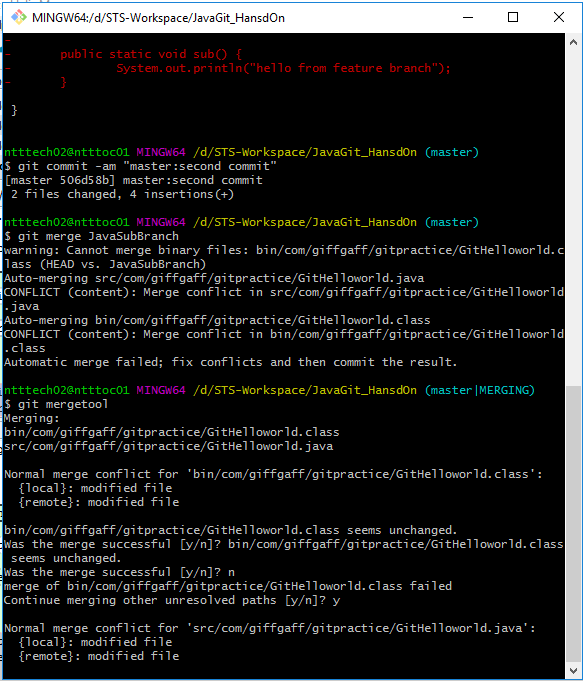
git config --global merge.tool p4merge

git config --global mergetool.p4merge.path 'C:\Program Files\Perforce\p4merge.exe'

Merge master and feature branch JavaSubBranch from Master using command

**git mergetool**

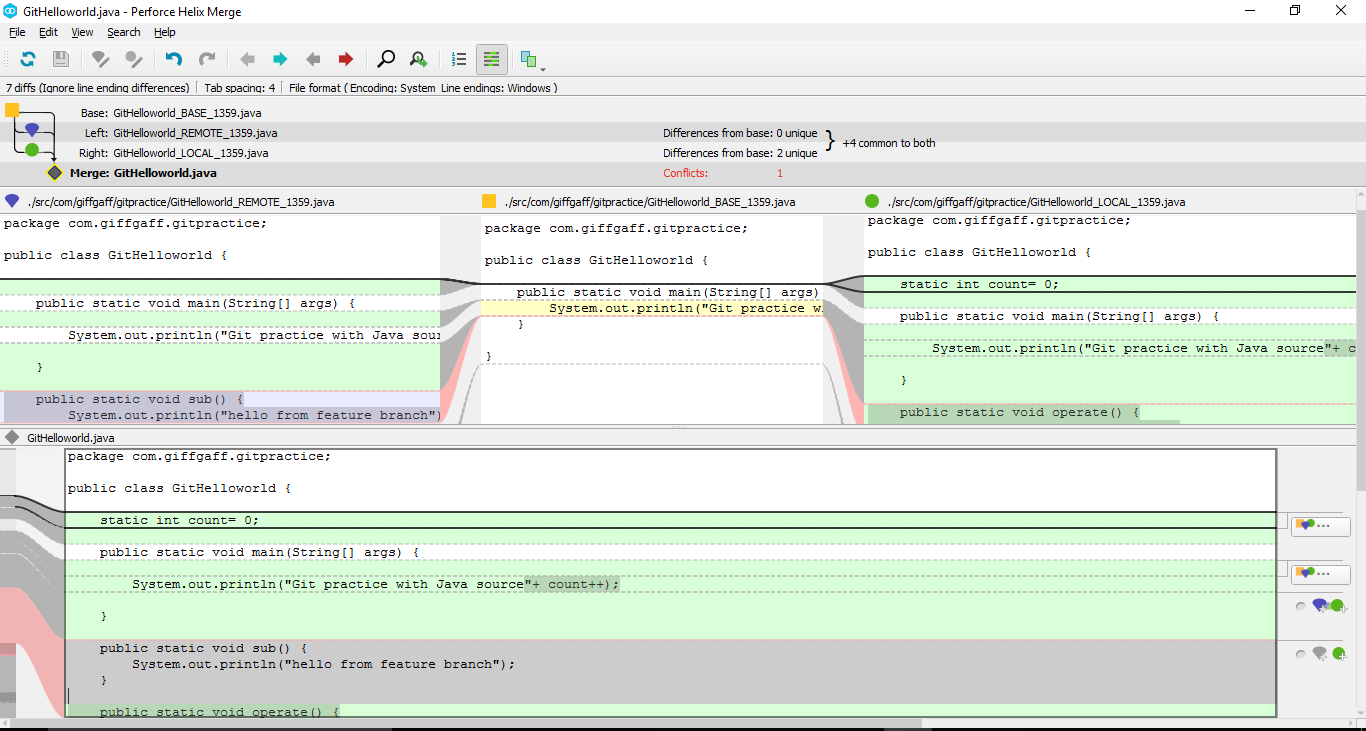
p4merge tool will be opened once for each conflicting file .



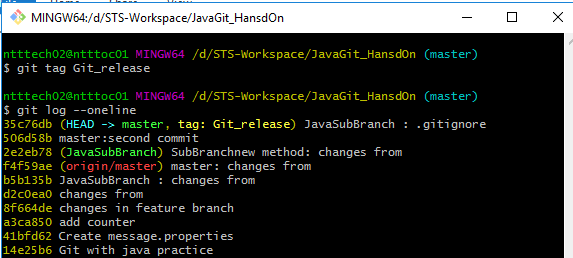
Here p4merge shows the conflicts between different Master(**REMOTE**), JavaSubBranch (**LOCAL**)

and **BASE** the ancestor of the 2 branches. The bottom window is used to resolve the conflicts.

Once resolved the file is saved and Exit p4merge

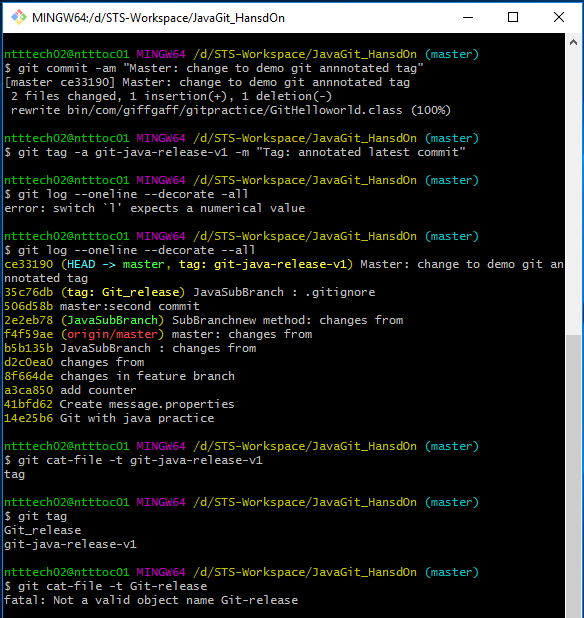


Create a light weight tag from latest commit of Master branch

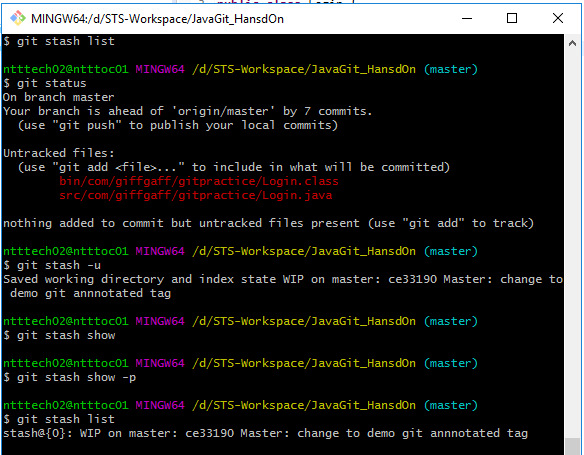


Make some changes and create an annotated on the latest commit.

And check the tag types



**Create stash of uncommitted changes of new or updated files**



**Create a new branch from specific stash**

