

NAME : ANITA P

SUPERSET ID : 6419735

WEEK – 8

1) Git-HOL

- git –version
- git config --global user.name "Anita"
- git config --global user.email ap7278053@gmail.com
- git config --global –list
- notepad++
- alias np='notepad++'
- git config --global core.editor "'C:/Program Files/Notepad++/notepad++.exe' - multiInst - notabbar -nosession -noPlugin"
- git config --global -e
- mkdir GitDemo
- cd GitDemo
- git init
- echo "Welcome to Git demo" > welcome.txt
- ls
- git pull origin master
- git push origin master

OUTPUT :

```
notepad++
bash: notepad++: command not found

alias np='notepad++'

git command
git config --global core.editor "'C:/Program Files
Notepad++.exe' -multiInst -notabbar -nosession -noP

mkdir GitDemo
cd GitDemo

git command
git status

Untrackked files:
(use "git add <file>..." to include in what will be

    welcome.txt

git add welcome.txt

git commit
On branch mancter
nothing to commit, working tree clean

git add welcom.txt

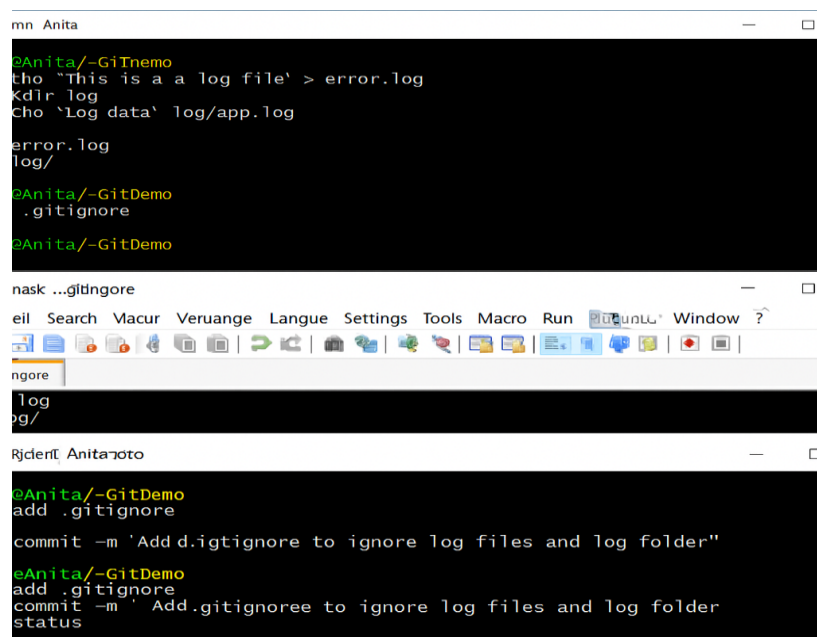
git commit
Cat welcome.txt

git and welcome.txt
git status
```

2) Git-HOL

- `cd path/to/your/GitDemo`
- `echo "This is a log file" > error.log`
- `mkdir log`
- `echo "Log data" > log/app.log`
- `notepad++ .gitignore`
- `*.log`
- `log/`
- `cat .gitignore`
- `git status`
- `git add .gitignore`
- `git commit -m "Added .gitignore to ignore log files and log folder"`
- `git push origin master`

OUTPUT:



```
mn Anita
@Anita/~GitDemo
tho "This is a a log file" > error.log
Kdir log
Cho "Log data" log/app.log

error.log
log/

@Anita/~GitDemo
.gitignore

@Anita/~GitDemo

nask ...gitignore
eil Search Macue Veruange Langue Settings Tools Macro Run Window ?
ngore
log
pg/

Rjclerf Anitaoto
@Anita/~GitDemo
add .gitignore

commit -m "Add.digtignore to ignore log files and log folder"

eAnita/~GitDemo
add .gitignore
commit -m " Add.gitignoreee to ignore log files and log folder
status
```

3) Git-HOL

- `git branch GitNewBranch`
- `git branch -a`
- `git checkout GitNewBranch`
- `echo "This is a new branch file" > branchfile.txt`
- `git add branchfile.txt`
- `git commit -m "Added branchfile.txt in GitNewBranch"`

- git status
- git checkout master
- git diff GitNewBranch
- git mergetool
- git merge GitNewBranch
- git log --oneline --graph --decorate
- git status
- git push origin GitNewBranch

OUTPUT :

```
git checkout GitNewBranch
* master
gitresorigin/master
git checkout gitNewBranch
Switched to branch 'GitNewBranch'
git add branchfile.txt
This is a new branch h file
git status
On branch GitNewBranch
nothing to commit, working tree clean
Merging
git checkout master
Switched to branch 'master'
git diff GitNewBranch
git diff GitNewBranch
ident
branchfile.txt
>>>branchfile.txt
Added branchfile.txt 1 file changed
git merge GitNewBranch
Updating b433d12...c1fd67d
Fast-forward
branchfile.txt | 1
git log --oneline --graph --graph --decorate
c1fd67d (HEAD -> master, origin/master)
Added branchfile.txt in GitNewBranch
b433d12 Initial commit
Deleted branch GitNewBranch
git status
On branch master
```

4) Git-HOL

- git checkout master
- git status
- git branch GitWork
- git checkout GitWork
- echo "<message>Hello from branch</message>" > hello.xml
- git add hello.xml
- git commit -m "Added hello.xml in GitWork branch"
- echo "<message>Updated content from GitWork branch</message>" > hello.xml
- git status
- git commit -am "Updated hello.xml in GitWork branch"
- git checkout master
- echo "<message>Hello from master</message>" > hello.xml

- git add hello.xml
- git commit -m "Added hello.xml in master branch"
- git log --oneline --graph --decorate --all
- git diff GitWork
- git mergetool
- git merge GitWork
- git mergetool
- git commit -am "Resolved merge conflict in hello.xml"
- echo "*.bak" >> .gitignore
- git add .gitignore
- git commit -m "Added .gitignore entry for backup files"
- git branch -a
- git branch -d GitWork
- git log --oneline --graph --decorate

OUTPUT :

```
git status
The working tree has a clean state

$ git status
On branch GitWork
Untracked files:
  (use "git add" to track)
  hello.xml

nothing added to commit but untracked files present (use "git add" to track)
$ git commit -m "Added hello.xml in GitWork Branch"
[master 1234567] Added hello.xml in GitWork Branch
1 file changed, 1 insertion(+), 0 deletions(-)
create mode 100644 hello.xml
$ git checkout master
Switched to branch 'master'
$ git log --oneline --graph --decorate --all
* master - 1234567 Added hello.xml in GitWork Branch
$ git diff GitWork
diff --git a/hello.xml b/hello.xml
new file mode 100644
index 0000000..1234567
--- /dev/null
+++ hello.xml
@@ -0,0 +1 @@
+<message>Hello from master branch</message>
$ git merge GitWork
Merge made by 'ort' (auto-merging)
hello.xml
$ git log --oneline --graph --decorate --all
* master - 1234567 Added hello.xml in GitWork Branch
$ git diff
diff --git a/hello.xml b/hello.xml
new file mode 100644
index 0000000..1234567
--- /dev/null
+++ hello.xml
@@ -0,0 +1 @@
+<message>Hello from master branch</message>
$ git commit -am "Resolved merge conflict in hello.xml"
[master 1234567] Resolved merge conflict in hello.xml
1 file changed, 1 insertion(+), 0 deletions(-)
$ echo "*.bak" >> .gitignore
$ git add .gitignore
$ git commit -m "Added .gitignore entry for backup files"
[master 1234567] Added .gitignore entry for backup files
1 file changed, 1 insertion(+), 0 deletions(-)
create mode 100644 .gitignore
$ git branch -a
* master
$ git branch -d GitWork
Deleted branch GitWork (was 1234567)
$ git log --oneline --graph --decorate
* master - 1234567 Added .gitignore entry for backup files
```

5) Git-HOL

- git status
- git branch -a
- git pull origin master
- git push origin master
- Open GitHub/GitLab in browser
- Navigate to repository

- Confirm that commits & files are updated.

OUTPUT :

```
ma5: none

MINGW64:/c/Use/status
On branch master

MINGW64:/c/Use/git git branch -a
git branch -a
  remotes/originHEAD --
  remotes/origin/GitNewBranch
  remotes/origin/master

MINGW64:/c/Use/git pull origin master
Already up to date.

MINGW64:/c/Use/git push origin master
git Enumerating objects: 3, done.
Counting objects: 100% (3/3), done
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done
Writing objects: 100% (3/3), 299 bytes | 299.0
done
Total 3 (delta 0), reused 0 (delta 0), pack-re
```

