**1) GIT hands on exercise 1**

git --version

# Output:

# git version 2.44.0

git config --global user.name "Aditya Anuragi"

git config --global user.email "aditya@gmail.com"

git config --list

# Output:

# user.name=Aditya Anuragi

# [user.email=aditya@gmail.com](mailto:user.email=aditya@gmail.com)

alias np="notepad++"

git config --global core.editor "notepad++ -multiInst -nosession"

git config --list

# Output

# core.editor=notepad++ -multiInst -nosession

git init GitDemo

# Output:

# Initialized empty Git repository in /learningGit/.git/

cd GitDemo

echo "Welcome to Git" > welcome.txt

git add welcome.txt

git commit -m "Added welcome.txt"

# Output:

# [master (root-commit) abc1234] Added welcome.txt

# 1 file changed, 1 insertion(+)

# create mode 100644 welcome.txt

git remote add origin https://github.com/user/GitDemo.git

git pull origin master

# Output:

# Already up to date.

git push origin master

# Output:

# Enumerating objects: 3, done.

# Counting objects: 100% (3/3), done.

# Writing objects: 100% (3/3), 245 bytes | 245.00 KiB/s, done.

# To https://github.com/user/GitDemo.git

# \* [new branch] master -> master

**2) GIT hands on exercise 2**

echo "Debug logs" > debug.log

mkdir log

touch log/test.txt

echo "\*.log" >> .gitignore

echo "log/" >> .gitignore

git add .gitignore

git commit -m "Ignore .log files and log folder"

# Output:

# [master 1234abc] Ignore .log files and log folder

# 1 file changed, 2 insertions(+)

# create mode 100644 .gitignore

git status

# Output:

# On branch master

# nothing to commit, working tree clean

**3) GIT hands on exercise 3**

git branch GitNewBranch

git checkout GitNewBranch

# Output:

# Switched to branch 'GitNewBranch'

touch file1.txt

echo "New branch content" > file1.txt

git add .

git commit -m "Changes in GitNewBranch"

# Output:

# [GitNewBranch abc5678] Changes in GitNewBranch

# 1 file changed, 1 insertion(+)

# create mode 100644 file1.txt

git status

# Output:

# On branch GitNewBranch

# nothing to commit, working tree clean

git checkout master

# Output:

# Switched to branch 'master'

git diff GitNewBranch

# Output:

# diff --git a/file1.txt b/file1.txt

# new file mode 100644

# index 0000000..d95f3ad

# --- /dev/null

# +++ b/file1.txt

# @@

# +New branch content

git merge GitNewBranch

# Output:

# Updating 1234567..abc5678

# Fast-forward

# file1.txt | 1 +

# 1 file changed, 1 insertion(+)

# create mode 100644 file1.txt

git log --oneline --graph --decorate

# Output:

# \* abc5678 (HEAD -> master, GitNewBranch) Changes in GitNewBranch

# \* 1234567 Initial commit

git branch -d GitNewBranch

# Output:

# Deleted branch GitNewBranch (was abc5678).

**4) GIT hands on exercise 4**

git branch GitWork

git checkout GitWork

# Output:

# Switched to branch 'GitWork'

echo "<msg>Branch version</msg>" > hello.xml

git add hello.xml

git commit -m "Branch hello.xml"

# Output:

# [GitWork 2345bcd] Branch hello.xml

# 1 file changed, 1 insertion(+)

# create mode 100644 hello.xml

git checkout master

# Output:

# Switched to branch 'master'

echo "<msg>Master version</msg>" > hello.xml

git add hello.xml

git commit -m "Master hello.xml"

# Output:

# [master 3456cde] Master hello.xml

# 1 file changed, 1 insertion(+)

# create mode 100644 hello.xml

git log --oneline --graph --decorate --all

# Shows output of all the branches with all the commits colour coded

git diff GitWork

# Output shows the difference between master and GitWork hello.xml

git merge GitWork

# Output:

# Auto-merging hello.xml

# CONFLICT (content): Merge conflict in hello.xml

# Automatic merge failed; fix conflicts and then commit the result.

# Resolve conflict manually, then:

git add hello.xml

git commit -m "Resolved merge conflict in hello.xml"

# Output:

# [master 4567def] Resolved merge conflict in hello.xml

echo "\*.bak" >> .gitignore

git add .gitignore

git commit -m "Ignore backup files"

# Output:

# [master 5678efg] Ignore backup files

git branch -d GitWork

# Output:

# Deleted branch GitWork (was 2345bcd).

git log --oneline --graph --decorate

# Output shows merged history

**5) GIT hands on exercise 5**

git status

# Output:

# On branch master

# nothing to commit, working tree clean

git branch -a

# Output:

# \* master

# remotes/origin/master

git pull origin master

# Output:

# Already up to date.

git push origin master

# Output:

# Everything up-to-date