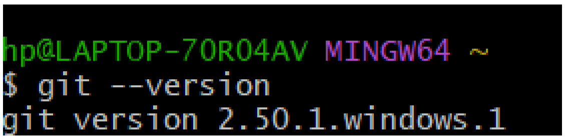
HandsOn 1:

Objectives -

Familiar with Git commands like git init, git status, git add, git commit, git push, and git pull.

Step 1: Setup your machine with Git Configuration

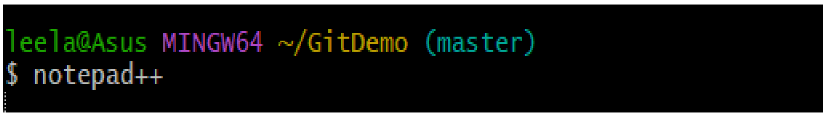
1) Knowing git version -



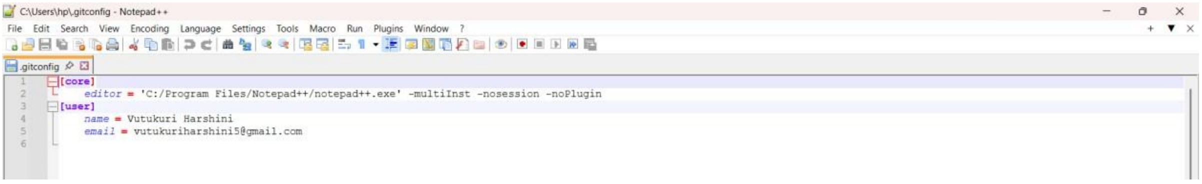
```
hp@LAPTOP-70R04AV MINGW64 ~
$ git --version
git version 2.50.1.windows.1
```

Step 2: Integrate notepad++.exe to Git and make it a default editor

Notepad++ is integrated and made it a default editor



```
leela@Asus MINGW64 ~/GitDemo (master)
$ notepad++
```

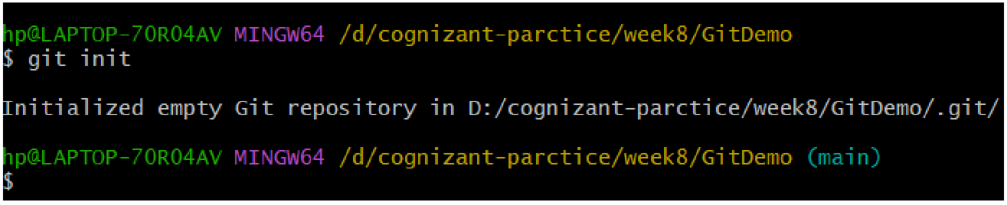


C:\Users\hp\.gitconfig - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
.gitconfig

```
[core]
    editor = 'C:/Program Files/Notepad++/notepad++.exe' -multiInst -nosession -noPlugin
[user]
    name = Vutukuri Harshini
    email = vutukuriharshini5@gmail.com
```

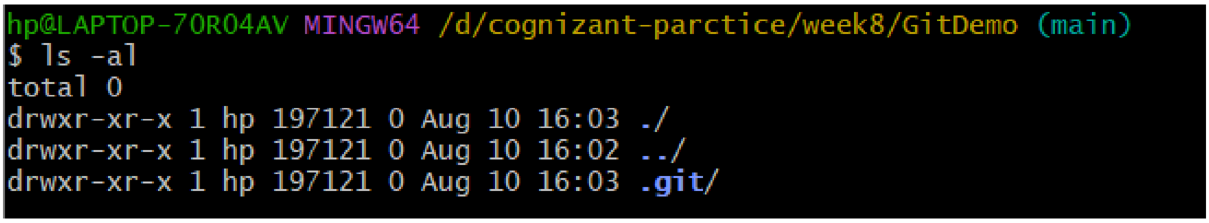
Step 3: Add a file to source code repository

1) Initialzing -



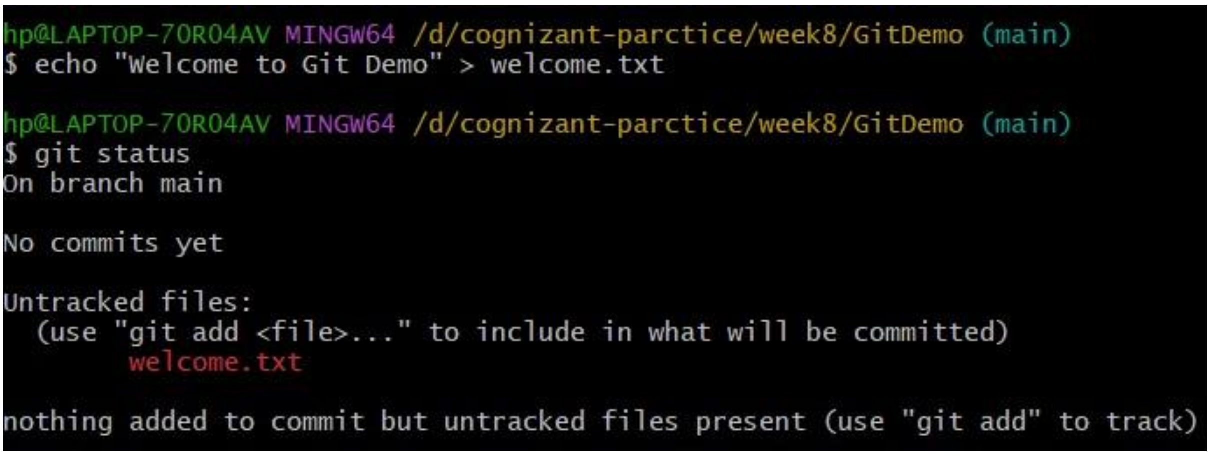
```
hp@LAPTOP-70R04AV MINGW64 /d/cognizant-parctice/week8/GitDemo
$ git init

Initialized empty Git repository in D:/cognizant-parctice/week8/GitDemo/.git/

hp@LAPTOP-70R04AV MINGW64 /d/cognizant-parctice/week8/GitDemo (main)
$
```

2) Cmd: ls -al



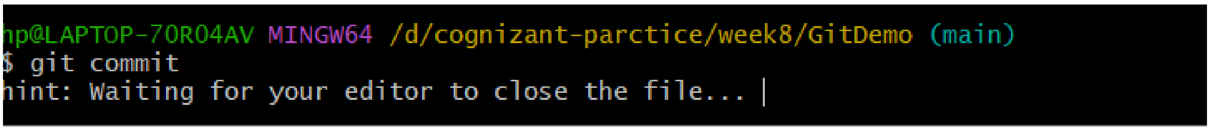
```
hp@LAPTOP-70R04AV MINGW64 /d/cognizant-parctice/week8/GitDemo (main)
$ ls -al
total 0
drwxr-xr-x 1 hp 197121 0 Aug 10 16:03 ./
drwxr-xr-x 1 hp 197121 0 Aug 10 16:02 ../
drwxr-xr-x 1 hp 197121 0 Aug 10 16:03 .git/
```

3) Verifying content &Checking git status -



```
hp@LAPTOP-70R04AV MINGW64 /d/cognizant-parctice/week8/GitDemo (main)
$ echo "Welcome to Git Demo" > welcome.txt

hp@LAPTOP-70R04AV MINGW64 /d/cognizant-parctice/week8/GitDemo (main)
$ git status
On branch main

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        welcome.txt

nothing added to commit but untracked files present (use "git add" to track)
```

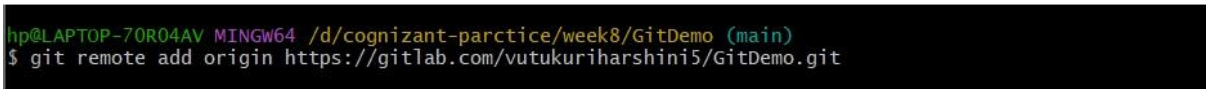
4) commit file



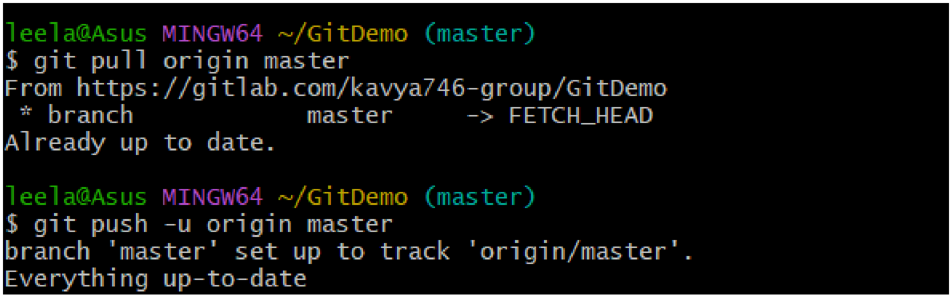
```
hp@LAPTOP-70R04AV MINGW64 /d/cognizant-parctice/week8/GitDemo (main)
$ git commit
hint: Waiting for your editor to close the file... |
```

5) Signed in with gitlab and created GitDemo repository -

URL: https://gitlab.com/vutukuriharshini5/GitDemo.git

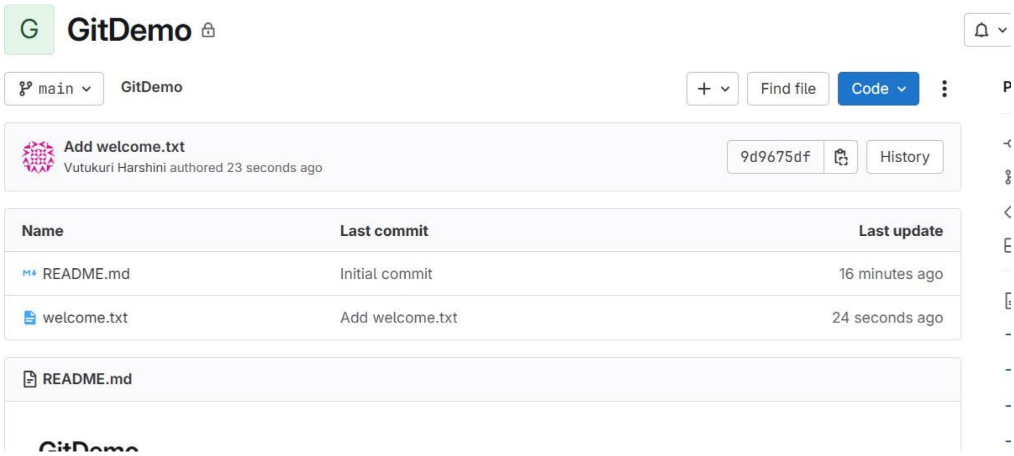


```
hp@LAPTOP-70R04AV MINGW64 /d/cognizant-parctice/week8/GitDemo (main)
$ git remote add origin https://gitlab.com/vutukuriharshini5/GitDemo.git
```



```
leela@Asus MINGW64 ~/GitDemo (master)
$ git pull origin master
From https://gitlab.com/kavya746-group/GitDemo
 * branch            master     -> FETCH_HEAD
Already up to date.

leela@Asus MINGW64 ~/GitDemo (master)
$ git push -u origin master
branch 'master' set up to track 'origin/master'.
Everything up-to-date
```

Output:



G GitDemo

main GitDemo

Find file Code

Add welcome.txt
Vutukuri Harshini authored 23 seconds ago

9d9675df History

Name	Last commit	Last update
README.md	Initial commit	16 minutes ago
welcome.txt	Add welcome.txt	24 seconds ago

README.md

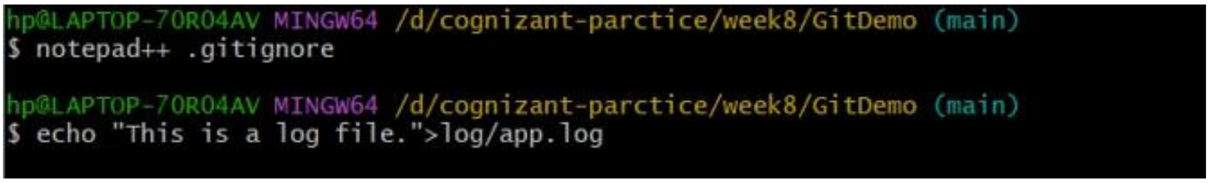
GitDemo

HandsOn 2:

Objectives -

Create a “.log” file and a log folder in the working directory of Git. Update the .gitignore file in such a way that on committing, these files (.log extensions and log folders) are ignored.

Step-1: Creating a log file

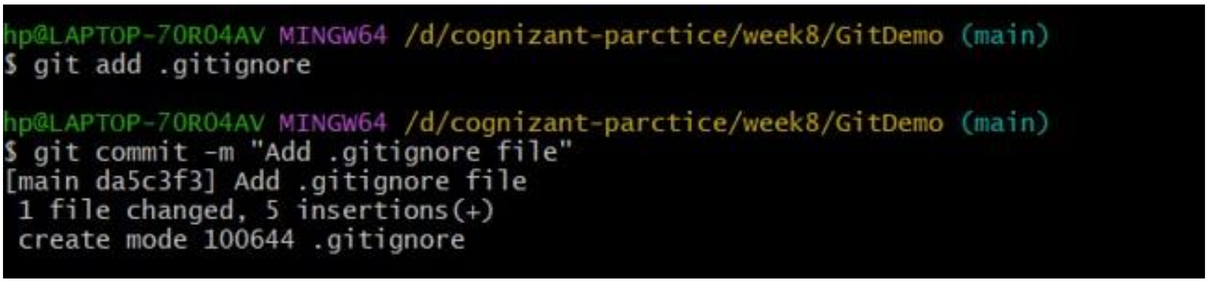


```
hp@LAPTOP-70R04AV MINGW64 /d/cognizant-parctice/week8/GitDemo (main)
$ notepad++ .gitignore

hp@LAPTOP-70R04AV MINGW64 /d/cognizant-parctice/week8/GitDemo (main)
$ echo "This is a log file.">log/app.log
```

Step-2: Creating .gitignore and adding rules,

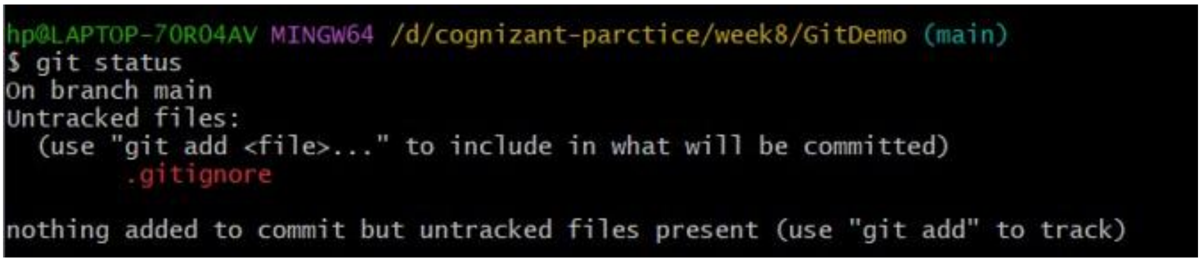
Step-3: Local Repository Status - Only .gitignore is tracked.

Step-4:Stage and commit .gitignore -



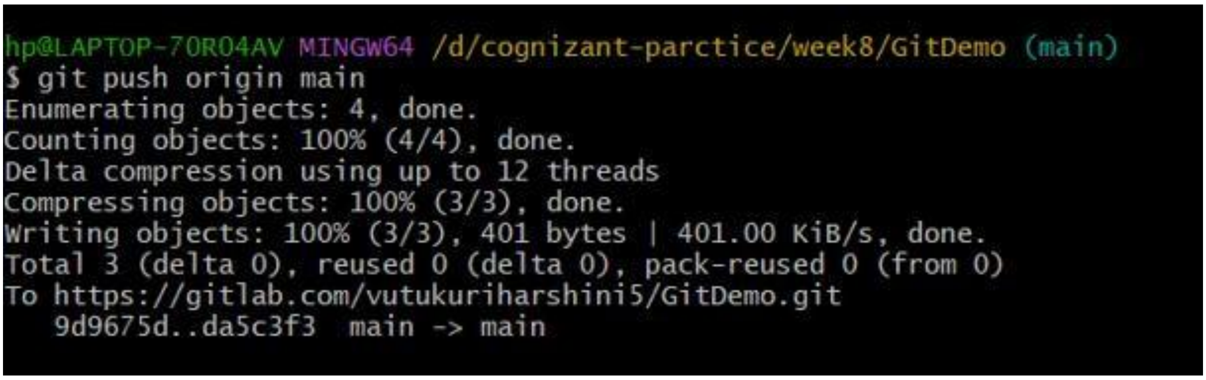
```
hp@LAPTOP-70R04AV MINGW64 /d/cognizant-parctice/week8/GitDemo (main)
$ git add .gitignore

hp@LAPTOP-70R04AV MINGW64 /d/cognizant-parctice/week8/GitDemo (main)
$ git commit -m "Add .gitignore file"
[main da5c3f3] Add .gitignore file
 1 file changed, 5 insertions(+)
 create mode 100644 .gitignore
```

Step-5: Verify with git status



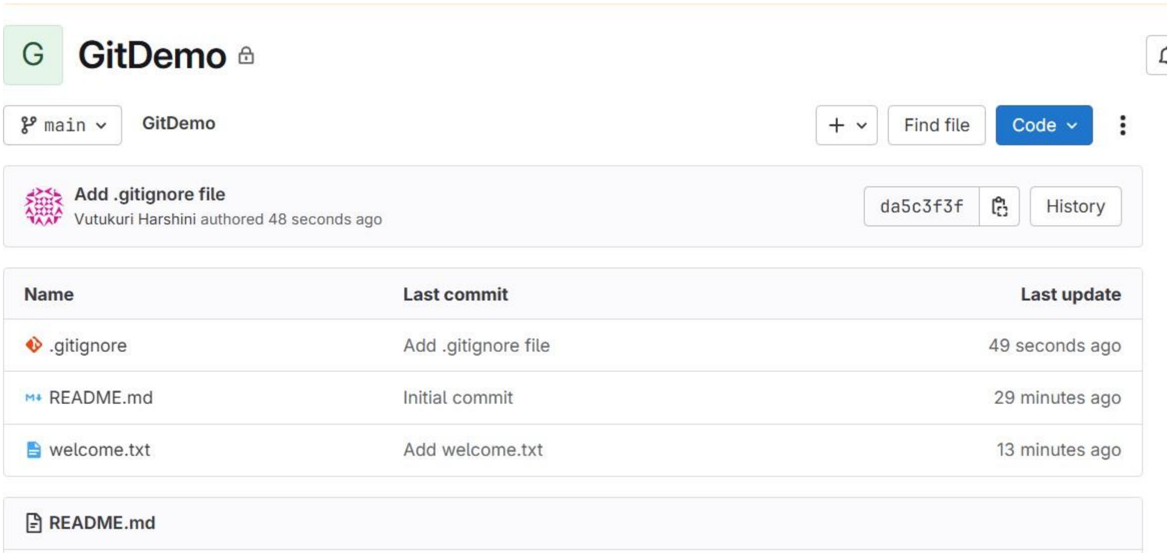
```
hp@LAPTOP-70R04AV MINGW64 /d/cognizant-parctice/week8/GitDemo (main)
$ git status
On branch main
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore

nothing added to commit but untracked files present (use "git add" to track)
```

Step-6: Push to remote



```
hp@LAPTOP-70R04AV MINGW64 /d/cognizant-parctice/week8/GitDemo (main)
$ git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 401 bytes | 401.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://gitlab.com/vutukuriharshini5/GitDemo.git
   9d9675d..da5c3f3  main -> main
```

Git Repository Status:



Add .gitignore file
Vutukuri Harshini authored 48 seconds ago

da5c3f3f

Name	Last commit	Last update
.gitignore	Add .gitignore file	49 seconds ago
README.md	Initial commit	29 minutes ago
welcome.txt	Add welcome.txt	13 minutes ago

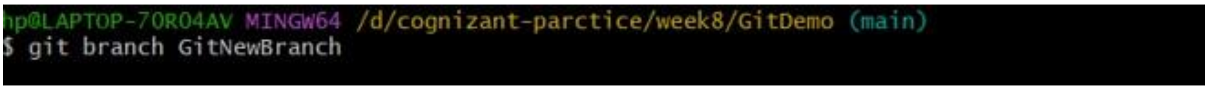
README.md

HandsOn 3:

Objective - Branching and Merging

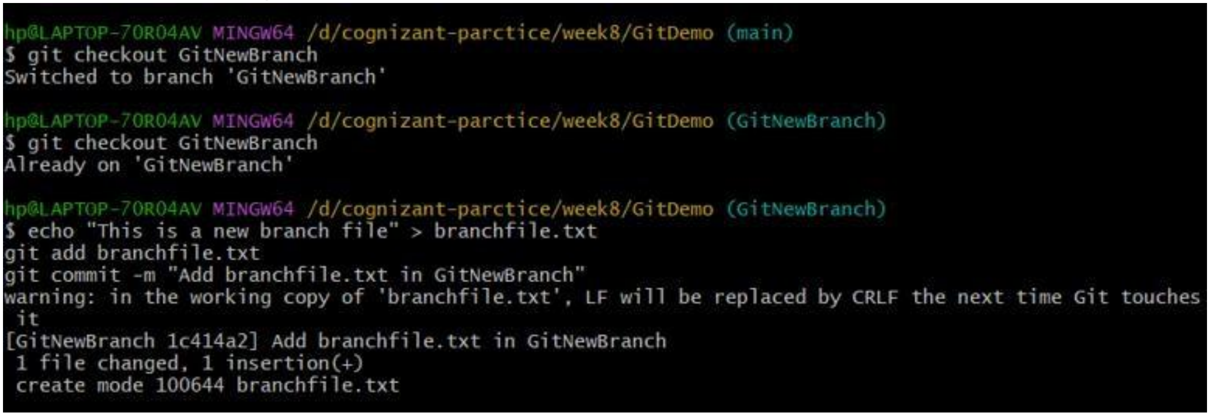
Branching:

1) Create a new branch “GitNewBranch”



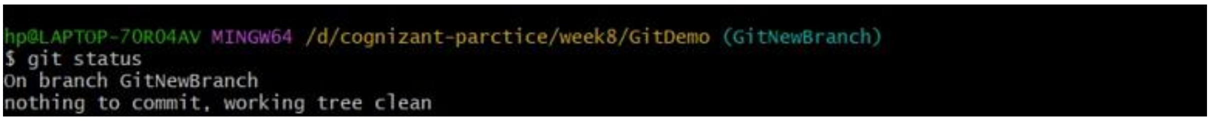
```
hp@LAPTOP-70R04AV MINGW64 /d/cognizant-parctice/week8/GitDemo (main)
$ git branch GitNewBranch
```

2) List all the local and remote branches
3) Switch to the newly created branch and add files
4) Commit the changes



```
hp@LAPTOP-70R04AV MINGW64 /d/cognizant-parctice/week8/GitDemo (main)
$ git checkout GitNewBranch
Switched to branch 'GitNewBranch'

hp@LAPTOP-70R04AV MINGW64 /d/cognizant-parctice/week8/GitDemo (GitNewBranch)
$ git checkout GitNewBranch
Already on 'GitNewBranch'

hp@LAPTOP-70R04AV MINGW64 /d/cognizant-parctice/week8/GitDemo (GitNewBranch)
$ echo "This is a new branch file" > branchfile.txt
git add branchfile.txt
git commit -m "Add branchfile.txt in GitNewBranch"
warning: in the working copy of 'branchfile.txt', LF will be replaced by CRLF the next time Git touches
 it
[GitNewBranch 1c414a2] Add branchfile.txt in GitNewBranch
 1 file changed, 1 insertion(+)
 create mode 100644 branchfile.txt
```

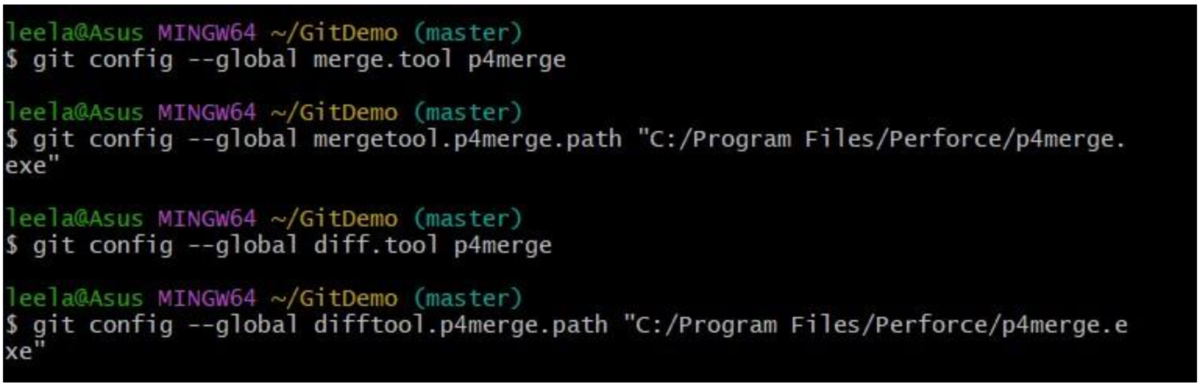
5) Check the status



```
hp@LAPTOP-70R04AV MINGW64 /d/cognizant-parctice/week8/GitDemo (GitNewBranch)
$ git status
On branch GitNewBranch
nothing to commit, working tree clean
```

Merging:

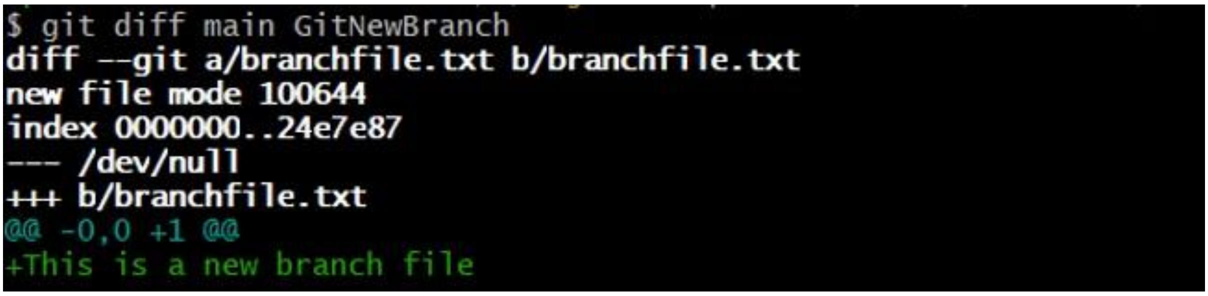
Commands used for Git to use P4Merge -



```
leela@Asus MINGW64 ~/GitDemo (master)
$ git config --global merge.tool p4merge

leela@Asus MINGW64 ~/GitDemo (master)
$ git config --global mergetool.p4merge.path "C:/Program Files/Perforce/p4merge.
exe"

leela@Asus MINGW64 ~/GitDemo (master)
$ git config --global diff.tool p4merge

leela@Asus MINGW64 ~/GitDemo (master)
$ git config --global difftool.p4merge.path "C:/Program Files/Perforce/p4merge.e
xe"
```

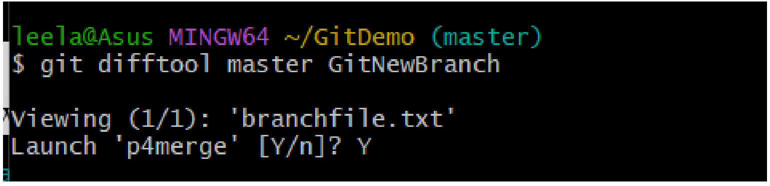
1) Switch to the master

2) List out all the differences between trunk and branch

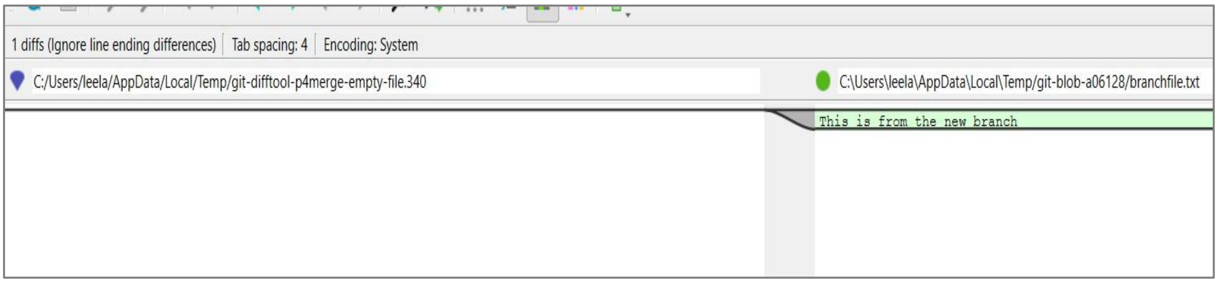


```
$ git diff main GitNewBranch
diff --git a/branchfile.txt b/branchfile.txt
new file mode 100644
index 0000000..24e7e87
--- /dev/null
+++ b/branchfile.txt
@@ -0,0 +1 @@
+This is a new branch file
```

3) List out all the visual differences between master and branch using **P4Merge tool**



```
leela@Asus MINGW64 ~/GitDemo (master)
$ git difftool master GitNewBranch

Viewing (1/1): 'branchfile.txt'
Launch 'p4merge' [Y/n]? Y
```



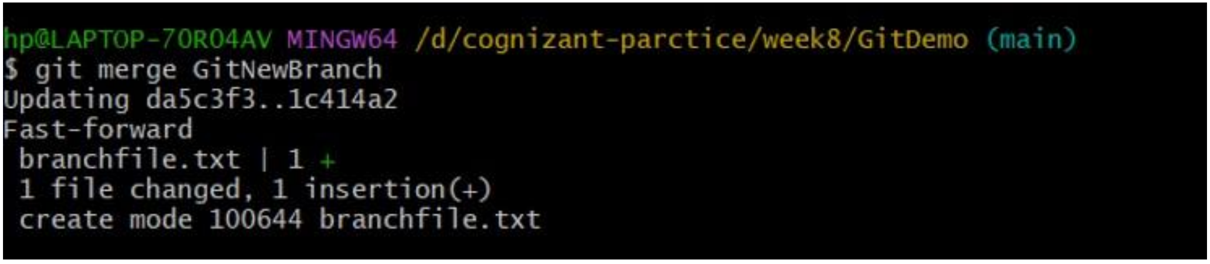
1 diffs (Ignore line ending differences) | Tab spacing: 4 | Encoding: System

C:/Users/leela/AppData/Local/Temp/git-difftool-p4merge-empty-file.340

C:\Users\leela\AppData\Local\Temp/git-blob-a06128/branchfile.txt

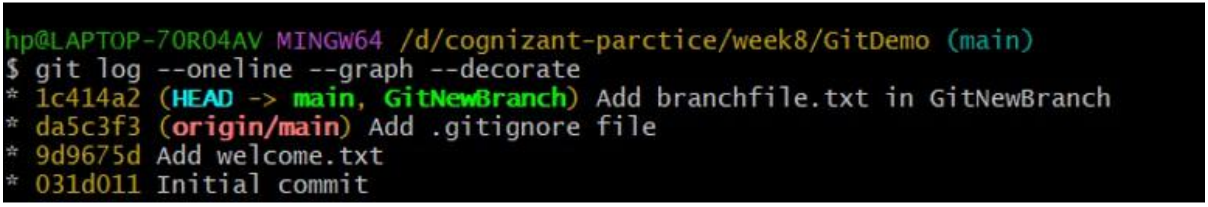
```
This is from the new branch
```

4) Merge the source branch to the trunk



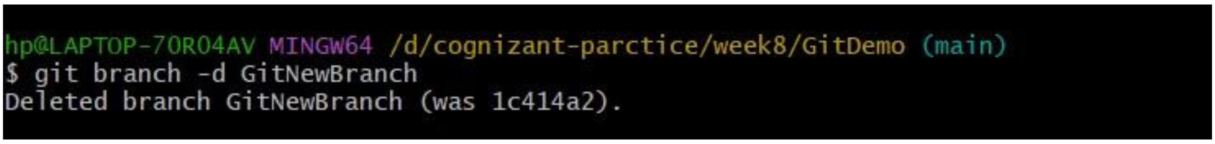
```
hp@LAPTOP-70R04AV MINGW64 /d/cognizant-parctice/week8/GitDemo (main)
$ git merge GitNewBranch
Updating da5c3f3..1c414a2
Fast-forward
 branchfile.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 branchfile.txt
```

5) Observe the logging after merging



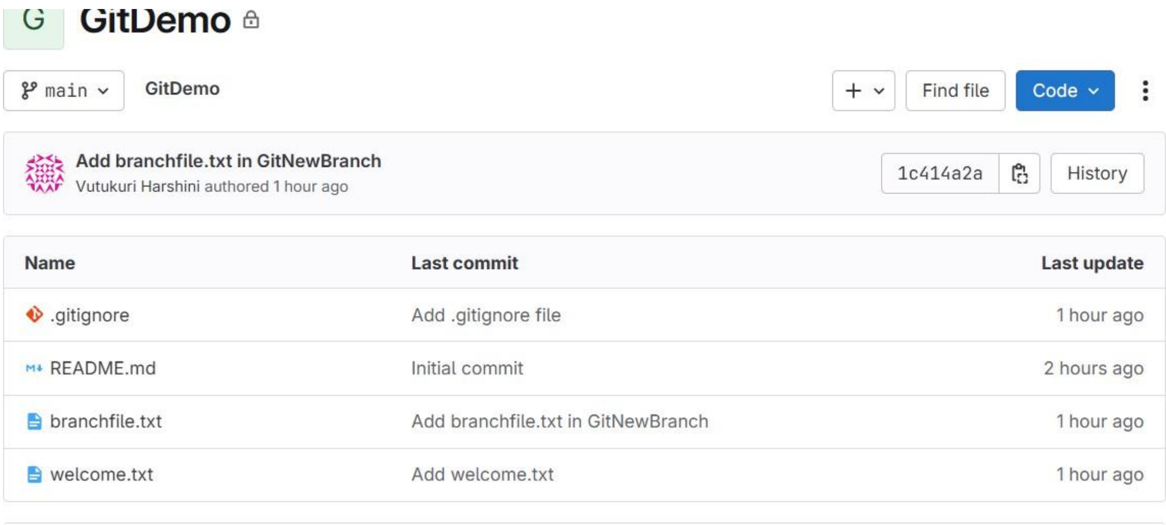
```
hp@LAPTOP-70R04AV MINGW64 /d/cognizant-parctice/week8/GitDemo (main)
$ git log --oneline --graph --decorate
* 1c414a2 (HEAD -> main, GitNewBranch) Add branchfile.txt in GitNewBranch
* da5c3f3 (origin/main) Add .gitignore file
* 9d9675d Add welcome.txt
* 031d011 Initial commit
```

6) Delete the branch after merging and observe status



```
hp@LAPTOP-70R04AV MINGW64 /d/cognizant-parctice/week8/GitDemo (main)
$ git branch -d GitNewBranch
Deleted branch GitNewBranch (was 1c414a2).
```

Output:



Add branchfile.txt in GitNewBranch

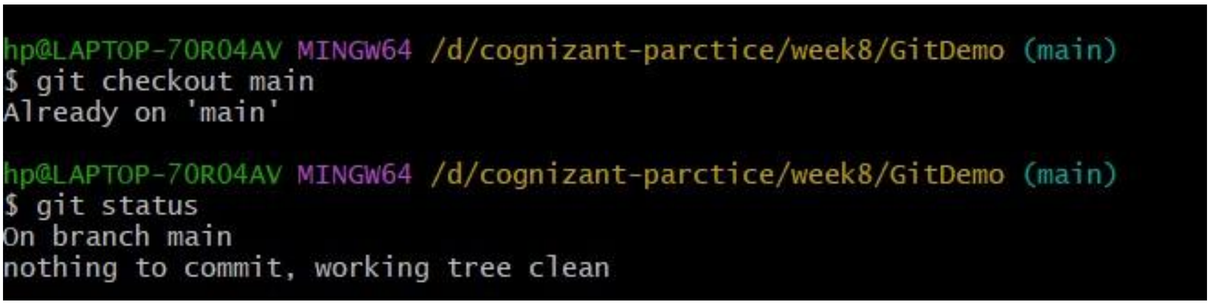
Vutukuri Harshini authored 1 hour ago

Name	Last commit	Last update
.gitignore	Add .gitignore file	1 hour ago
README.md	Initial commit	2 hours ago
branchfile.txt	Add branchfile.txt in GitNewBranch	1 hour ago
welcome.txt	Add welcome.txt	1 hour ago

HandsOn 4:

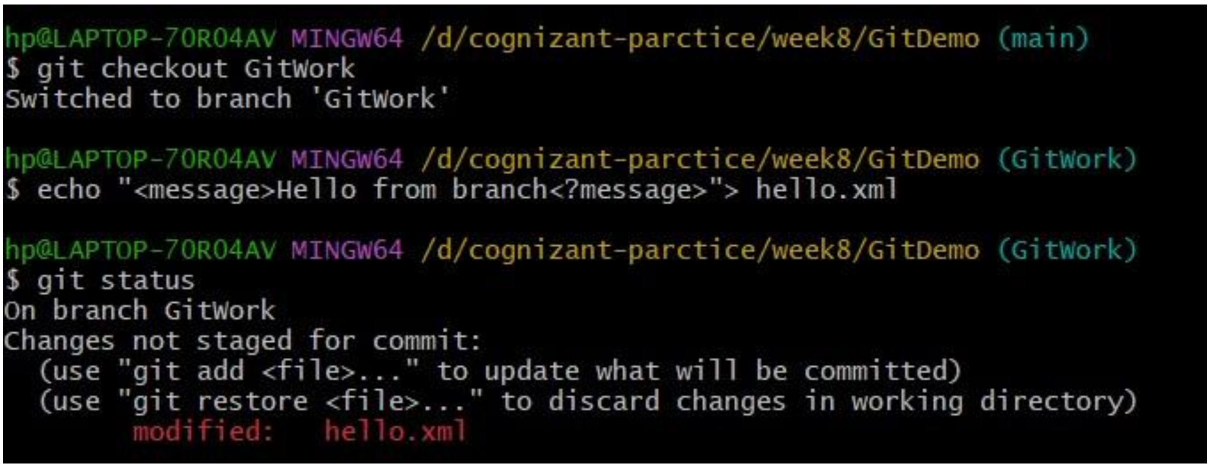
Objective - Explain how to resolve the conflict during merge

1) Verify master is clean



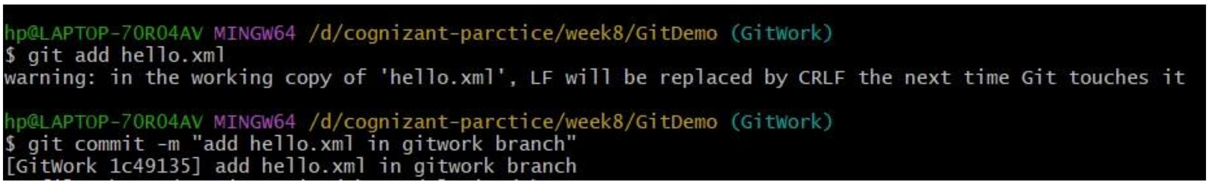
```
hp@LAPTOP-70R04AV MINGW64 /d/cognizant-parctice/week8/GitDemo (main)
$ git checkout main
Already on 'main'

hp@LAPTOP-70R04AV MINGW64 /d/cognizant-parctice/week8/GitDemo (main)
$ git status
On branch main
nothing to commit, working tree clean
```

2) Create branch and add file



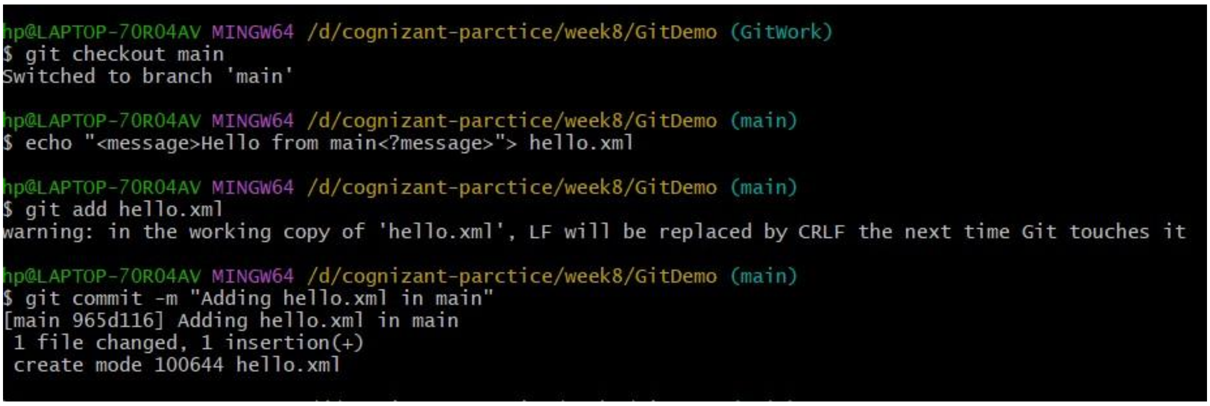
```
hp@LAPTOP-70R04AV MINGW64 /d/cognizant-parctice/week8/GitDemo (main)
$ git checkout GitWork
Switched to branch 'GitWork'

hp@LAPTOP-70R04AV MINGW64 /d/cognizant-parctice/week8/GitDemo (GitWork)
$ echo "<message>Hello from branch<?message>"> hello.xml

hp@LAPTOP-70R04AV MINGW64 /d/cognizant-parctice/week8/GitDemo (GitWork)
$ git status
On branch GitWork
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   hello.xml
```

3) Commit changes in branch



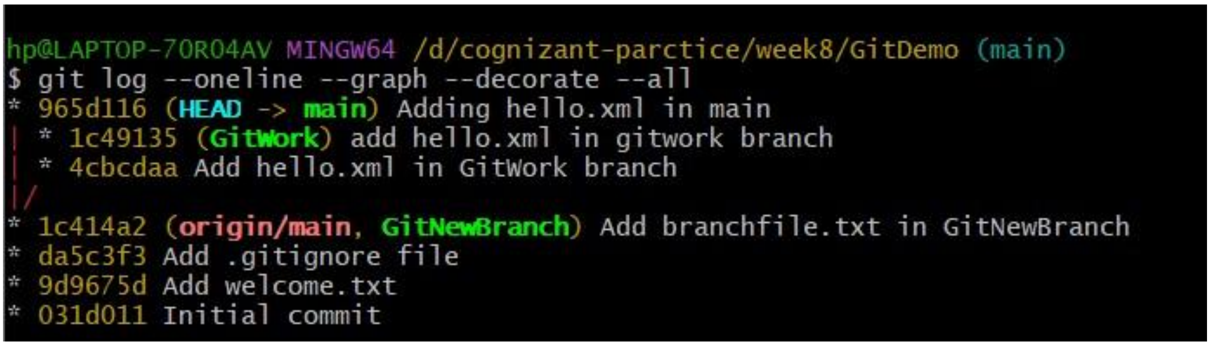
```
hp@LAPTOP-70R04AV MINGW64 /d/cognizant-parctice/week8/GitDemo (GitWork)
$ git add hello.xml
warning: in the working copy of 'hello.xml', LF will be replaced by CRLF the next time Git touches it

hp@LAPTOP-70R04AV MINGW64 /d/cognizant-parctice/week8/GitDemo (GitWork)
$ git commit -m "add hello.xml in gitwork branch"
[GitWork 1c49135] add hello.xml in gitwork branch
```

4) Switch to master and create conflicting file



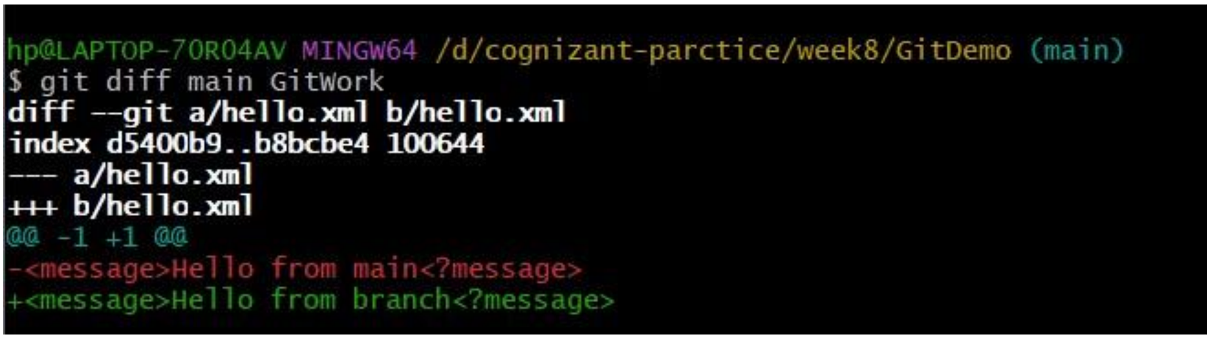
```
hp@LAPTOP-70R04AV MINGW64 /d/cognizant-parctice/week8/GitDemo (GitWork)
$ git checkout main
Switched to branch 'main'

hp@LAPTOP-70R04AV MINGW64 /d/cognizant-parctice/week8/GitDemo (main)
$ echo "<message>Hello from main<?message>"> hello.xml

hp@LAPTOP-70R04AV MINGW64 /d/cognizant-parctice/week8/GitDemo (main)
$ git add hello.xml
warning: in the working copy of 'hello.xml', LF will be replaced by CRLF the next time Git touches it

hp@LAPTOP-70R04AV MINGW64 /d/cognizant-parctice/week8/GitDemo (main)
$ git commit -m "Adding hello.xml in main"
[main 965d116] Adding hello.xml in main
 1 file changed, 1 insertion(+)
 create mode 100644 hello.xml
```

5) View logs



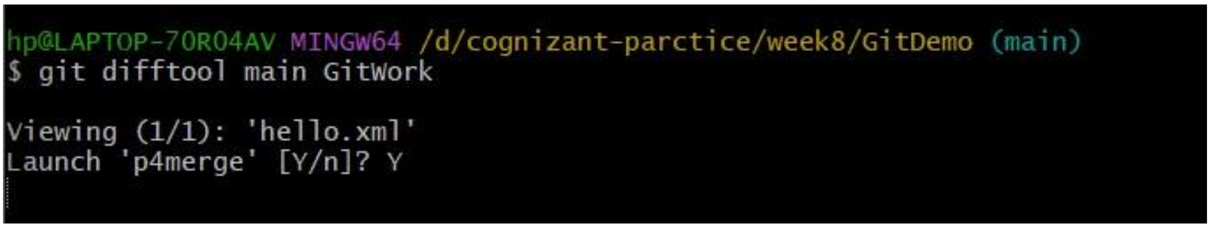
```
hp@LAPTOP-70R04AV MINGW64 /d/cognizant-parctice/week8/GitDemo (main)
$ git log --oneline --graph --decorate --all
* 965d116 (HEAD -> main) Adding hello.xml in main
| * 1c49135 (GitWork) add hello.xml in gitwork branch
| * 4cbcdaa Add hello.xml in GitWork branch
|/
* 1c414a2 (origin/main, GitNewBranch) Add branchfile.txt in GitNewBranch
* da5c3f3 Add .gitignore file
* 9d9675d Add welcome.txt
* 031d011 Initial commit
```

6) Check differences in command-line

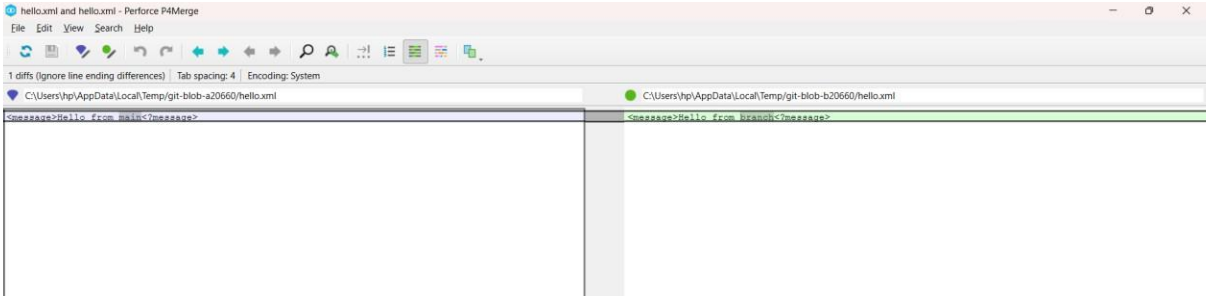


```
hp@LAPTOP-70R04AV MINGW64 /d/cognizant-parctice/week8/GitDemo (main)
$ git diff main GitWork
diff --git a/hello.xml b/hello.xml
index d5400b9..b8bcbe4 100644
--- a/hello.xml
+++ b/hello.xml
@@ -1 +1 @@
-<message>Hello from main<?message>
+<message>Hello from branch<?message>
```

7) View differences in P4Merge



```
hp@LAPTOP-70R04AV MINGW64 /d/cognizant-parctice/week8/GitDemo (main)
$ git difftool main GitWork

Viewing (1/1): 'hello.xml'
Launch 'p4merge' [Y/n]? Y
```



1 diffs (Ignore line ending differences) | Tab spacing: 4 | Encoding: System

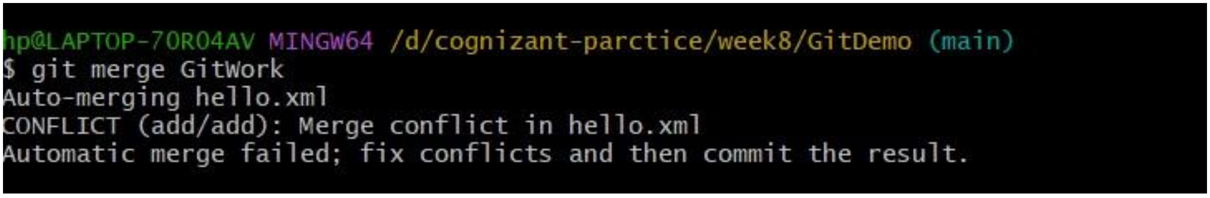
C:\Users\hp\AppData\Local\Temp/git-blob-a20660/hello.xml

```
<message>Hello from main<?message>
```

C:\Users\hp\AppData\Local\Temp/git-blob-b20660/hello.xml

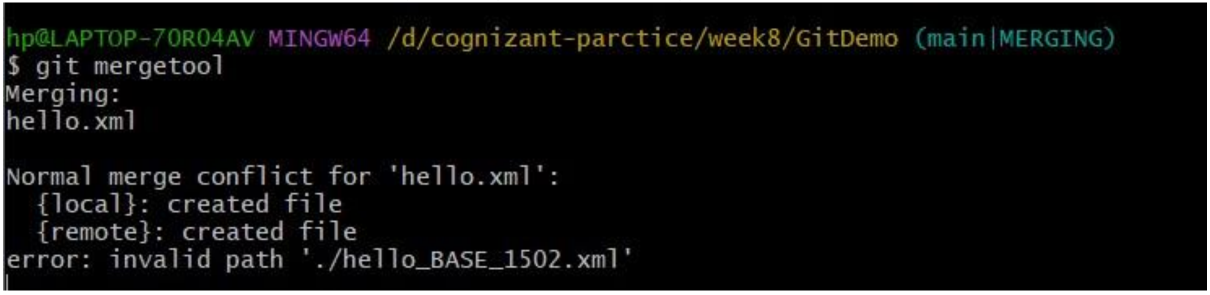
```
<message>Hello from branch<?message>
```

8) Attempt merge - cause conflict

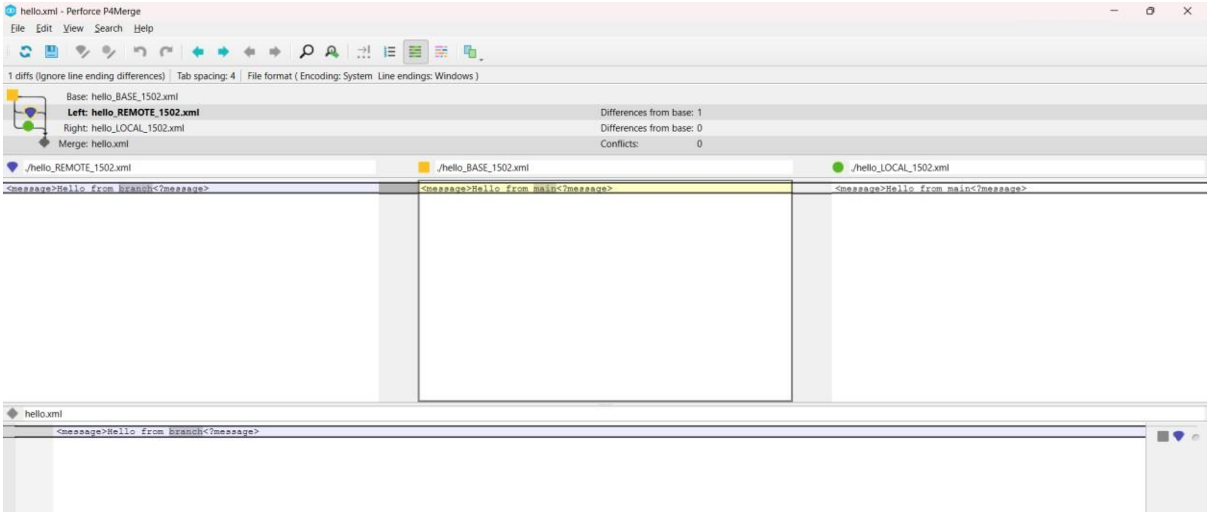


```
hp@LAPTOP-70R04AV MINGW64 /d/cognizant-parctice/week8/GitDemo (main)
$ git merge GitWork
Auto-merging hello.xml
CONFLICT (add/add): Merge conflict in hello.xml
Automatic merge failed; fix conflicts and then commit the result.
```

9) Resolve conflict with 3-way merge in P4Merge



```
hp@LAPTOP-70R04AV MINGW64 /d/cognizant-parctice/week8/GitDemo (main|MERGING)
$ git mergetool
Merging:
hello.xml

Normal merge conflict for 'hello.xml':
  {local}: created file
  {remote}: created file
error: invalid path './hello_BASE_1502.xml'
```



1 diffs (Ignore line ending differences) | Tab spacing: 4 | File format (Encoding: System Line endings: Windows)

Base: hello_BASE_1502.xml
Left: hello_REMOTE_1502.xml Differences from base: 1
Right: hello_LOCAL_1502.xml Differences from base: 0
Merge: hello.xml Conflicts: 0

./hello_REMOTE_1502.xml

```
<message>Hello from branch<?message>
```

./hello_BASE_1502.xml

```
<message>Hello from main<?message>
```

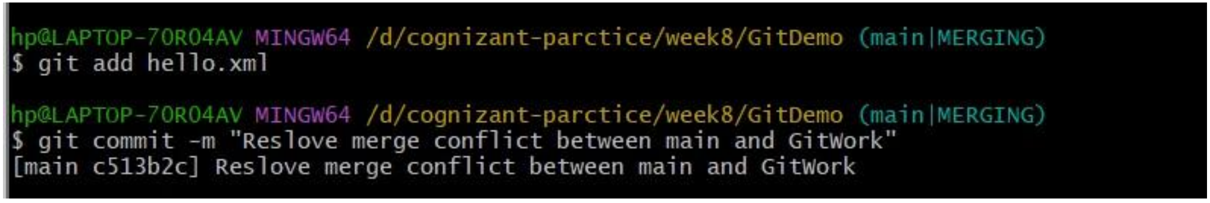
./hello_LOCAL_1502.xml

```
<message>Hello from main<?message>
```

hello.xml

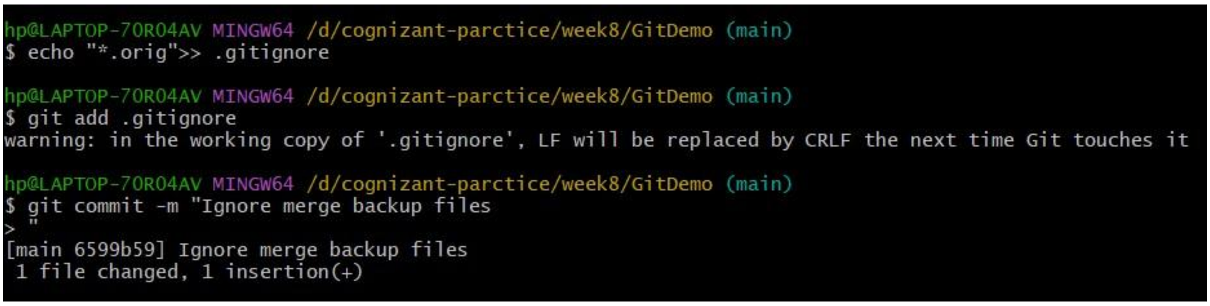
```
<message>Hello from branch<?message>
```

10) Mark conflict as resolved and commit



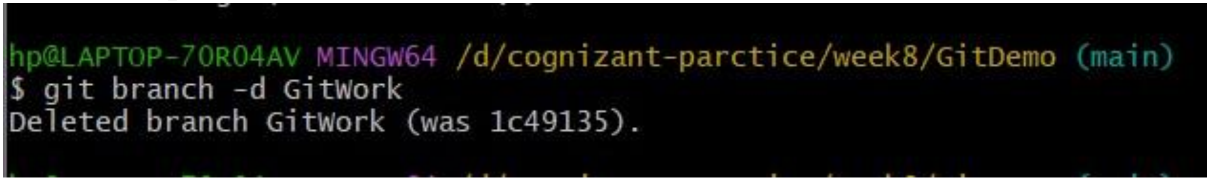
```
hp@LAPTOP-70R04AV MINGW64 /d/cognizant-parctice/week8/GitDemo (main|MERGING)
$ git add hello.xml

hp@LAPTOP-70R04AV MINGW64 /d/cognizant-parctice/week8/GitDemo (main|MERGING)
$ git commit -m "Reslove merge conflict between main and GitWork"
[main c513b2c] Reslove merge conflict between main and GitWork
```

11) Add backup file from merge to .gitignore



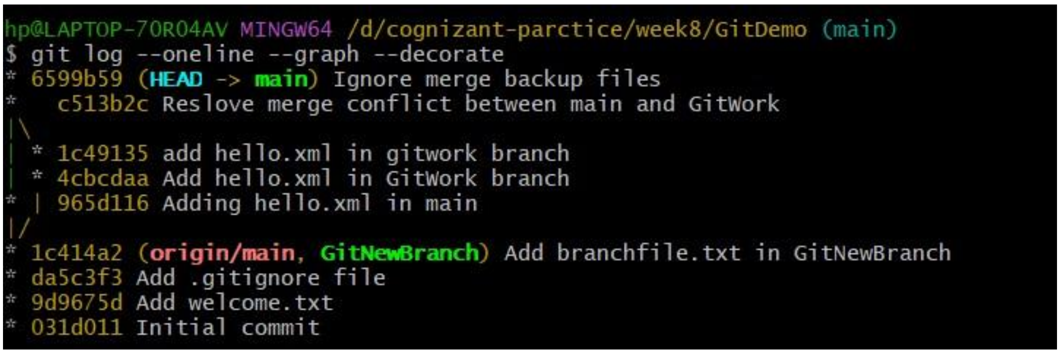
```
hp@LAPTOP-70R04AV MINGW64 /d/cognizant-parctice/week8/GitDemo (main)
$ echo "*.orig">> .gitignore

hp@LAPTOP-70R04AV MINGW64 /d/cognizant-parctice/week8/GitDemo (main)
$ git add .gitignore
warning: in the working copy of '.gitignore', LF will be replaced by CRLF the next time Git touches it

hp@LAPTOP-70R04AV MINGW64 /d/cognizant-parctice/week8/GitDemo (main)
$ git commit -m "Ignore merge backup files
> "
[main 6599b59] Ignore merge backup files
 1 file changed, 1 insertion(+)
```

12) Cleanup branch



```
hp@LAPTOP-70R04AV MINGW64 /d/cognizant-parctice/week8/GitDemo (main)
$ git branch -d GitWork
Deleted branch GitWork (was 1c49135).
```

13) Final log view

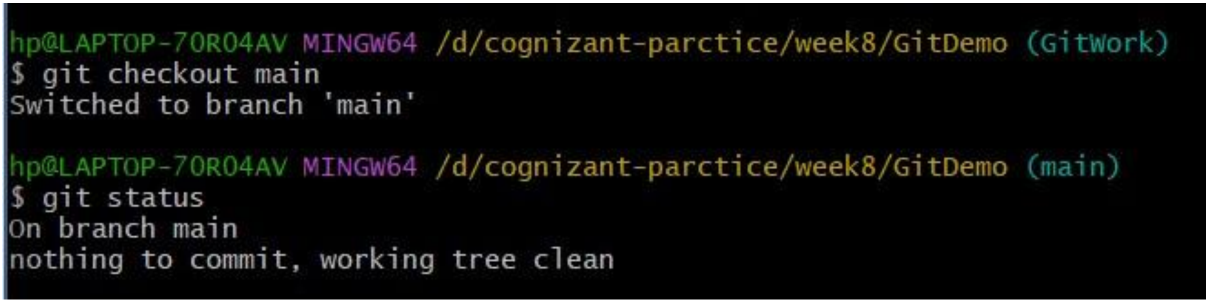


```
hp@LAPTOP-70R04AV MINGW64 /d/cognizant-parctice/week8/GitDemo (main)
$ git log --oneline --graph --decorate
* 6599b59 (HEAD -> main) Ignore merge backup files
*   c513b2c Reslove merge conflict between main and GitWork
|\
| * 1c49135 add hello.xml in gitwork branch
| * 4cbcdaa Add hello.xml in GitWork branch
* | 965d116 Adding hello.xml in main
|/
* 1c414a2 (origin/main, GitNewBranch) Add branchfile.txt in GitNewBranch
* da5c3f3 Add .gitignore file
* 9d9675d Add welcome.txt
* 031d011 Initial commit
```

HandsOn 5:

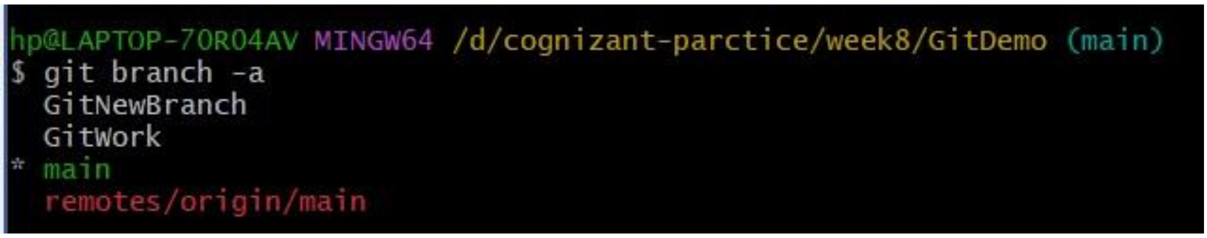
Objective - Explain how to clean up and push back to remote Git

1) Verify master is clean



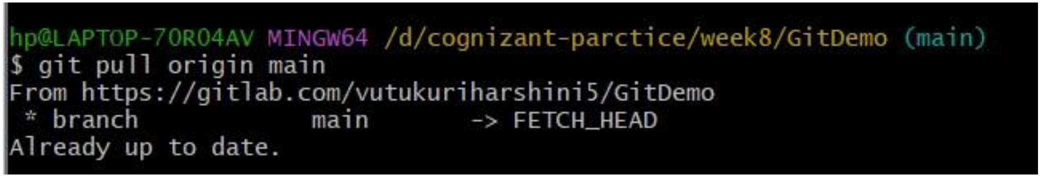
```
hp@LAPTOP-70R04AV MINGW64 /d/cognizant-parctice/week8/GitDemo (GitWork)
$ git checkout main
Switched to branch 'main'

hp@LAPTOP-70R04AV MINGW64 /d/cognizant-parctice/week8/GitDemo (main)
$ git status
On branch main
nothing to commit, working tree clean
```

2) List all branches



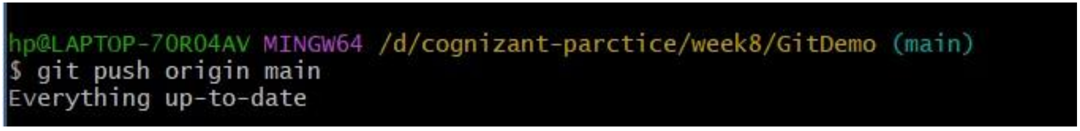
```
hp@LAPTOP-70R04AV MINGW64 /d/cognizant-parctice/week8/GitDemo (main)
$ git branch -a
  GitNewBranch
  GitWork
* main
  remotes/origin/main
```

3) Pull latest changes from remote



```
hp@LAPTOP-70R04AV MINGW64 /d/cognizant-parctice/week8/GitDemo (main)
$ git pull origin main
From https://gitlab.com/vutukuriharshini5/GitDemo
 * branch            main       -> FETCH_HEAD
Already up to date.
```

4) Push pending changes from Git-T03-HOL_002



```
hp@LAPTOP-70R04AV MINGW64 /d/cognizant-parctice/week8/GitDemo (main)
$ git push origin main
Everything up-to-date
```