

GIT - Handson 4

1. Verifying if maste is in clean state

```
brave@Thiru?? MINGW64 /s/GitDemo (master)
$ git status
On branch master
Your branch is ahead of 'origin/master' by 3 commits.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```

2. Created a branch "GitWork" & Added a file "hello.xml"

```
brave@Thiru?? MINGW64 /s/GitDemo (master)
$ git checkout -b GitWork
Switched to a new branch 'GitWork'

brave@Thiru?? MINGW64 /s/GitDemo (GitWork)
$ echo "<greeting>Hello from GitWork branch</greeting>" > hello.xml
```

3. Updated the content of "hello.xml"

```
brave@Thiru?? MINGW64 /s/GitDemo (GitWork)
$ echo "<greeting>Hello from GitWork branch</greeting>" > hello.xml

brave@Thiru?? MINGW64 /s/GitDemo (GitWork)
$ git add hello.xml
warning: in the working copy of 'hello.xml', LF will be replaced by CRLF the nex
t time Git touches it
```

4. Commit the changes to reflect in the branch

```
brave@Thiru?? MINGW64 /s/GitDemo (GitWork)
$ git commit
[GitWork 0ec91e4] Added hello.xml in GitWork branch
1 file changed, 1 insertion(+)
create mode 100644 hello.xml
```

5. Switch to master

```
brave@Thiru?? MINGW64 /s/GitDemo (GitWork)
$ git checkout master
Switched to branch 'master'
Your branch is ahead of 'origin/master' by 3 commits.
  (use "git push" to publish your local commits)
```

6. Add a file **“hello.xml”** to the master and add some different content than previous

```
brave@Thiru?? MINGW64 /s/GitDemo (master)
$ echo "<greeting>Hello from master branch</greeting>" > hello.xml
```

7. Commit the changes to the master

```
brave@Thiru?? MINGW64 /s/GitDemo (master)
$ git commit
[master 2e986c4] Added hello.xml in master branch
1 file changed, 1 insertion(+)
create mode 100644 hello.xml
```

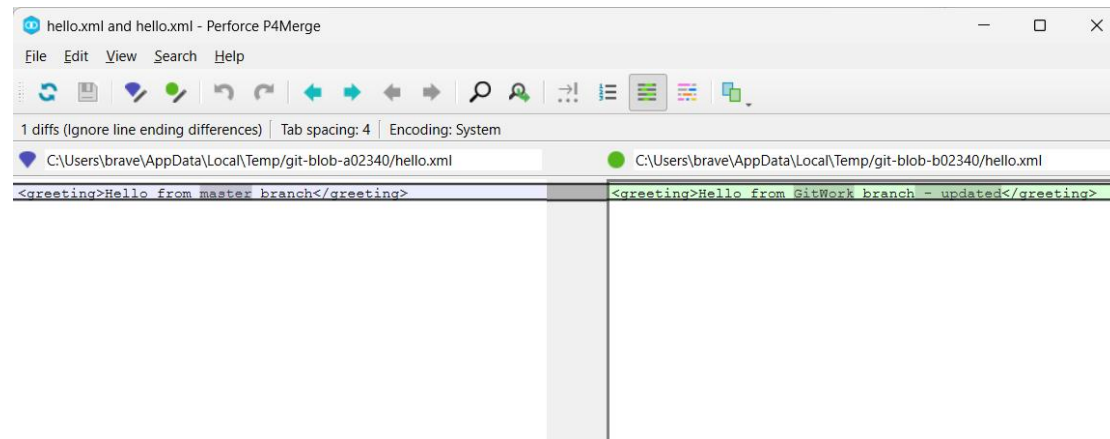
8. Observe the log by executing **“git log --oneline --graph --decorate --all”**

```
brave@Thiru?? MINGW64 /s/GitDemo (master)
$ git log --oneline --graph --decorate --all
* 2e986c4 (HEAD -> master) Added hello.xml in master branch
| * 2e374af (GitWork) Updated hello.xml in GitWork branch
| * 0ec91e4 Added hello.xml in GitWork branch
|/
| * bdfad9f Merge branch 'GitNewBranch'
|/
| * 949a89e Added newfile.txt in GitNewBranch
| * ec498f6 Add newfile.txt to master
|/
* 32807bf (origin/master) Added .gitignore to ignore .log files & log folders
* afe80b1 Added welcome.txt file
```

9. Checked the differences with Git diff tool

```
brave@Thiru?? MINGW64 /s/GitDemo (master)
$ git diff master GitWork
diff --git a/hello.xml b/hello.xml
index 2d8d754..437ce01 100644
--- a/hello.xml
+++ b/hello.xml
@@ -1,1 @@
- <greeting>Hello from master branch</greeting>
+ <greeting>Hello from GitWork branch - updated</greeting>
```

10. For better visualization, use P4Merge tool to list out all the differences between master and branch



11. Merged the branch to the master & Observe the git mark up

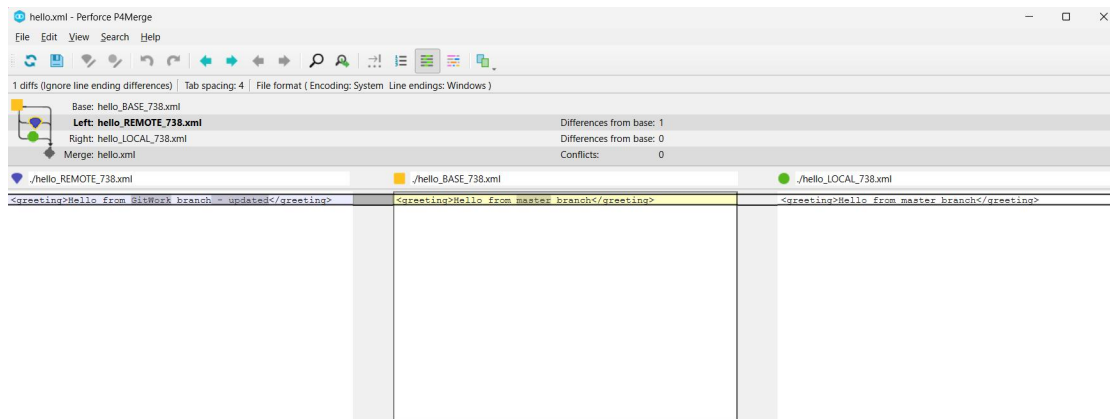
```
brave@Thiru?? MINGW64 /s/GitDemo (master)
$ git merge GitWork
Auto-merging hello.xml
CONFLICT (add/add): Merge conflict in hello.xml
Automatic merge failed; fix conflicts and then commit the result.
```

12. Use 3-way merge tool to resolve the conflict

```
brave@Thiru?? MINGW64 /s/GitDemo (master|MERGING)
$ git mergetool

This message is displayed because 'merge.tool' is not configured.
See 'git mergetool --tool-help' or 'git help config' for more details.
'git mergetool' will now attempt to use one of the following tools:
opendiff kdiff3 tkdiff xxdiff meld tortoisemerge gvimdiff diffuse diffmerge ecme
rge p4merge araxis bc codecompare smerge emerge vimdiff nvimdiff
Merging:
hello.xml

Normal merge conflict for 'hello.xml':
  {local}: created file
  {remote}: created file
Hit return to start merge resolution tool (p4merge):
```



13. Commit the changes to the master, once done with conflict

```
brave@Thiru?? MINGW64 /s/GitDemo (master|MERGING)
$ git add hello.xml

brave@Thiru?? MINGW64 /s/GitDemo (master|MERGING)
$ git commit
[master f2ba564] Merged branch 'GitWork'
```

14. Observe the git status and add backup file to the .gitignore file

```
brave@Thiru?? MINGW64 /s/GitDemo (master)
$ echo "*.orig" >> .gitignore
```

15. Commit the changes to the .gitignore

```
brave@Thiru?? MINGW64 /s/GitDemo (master)
$ git add .gitignore
warning: in the working copy of '.gitignore', LF will be replaced by CRLF the next time Git touches it

brave@Thiru?? MINGW64 /s/GitDemo (master)
$ git commit
[master 64cafd8] Ignore merge backup files
1 file changed, 1 insertion(+), 1 deletion(-)
```

16. List out all the available branches

```
brave@Thiru?? MINGW64 /s/GitDemo (master)
$ git branch
  GitWork
* master
```

17. Delete the branch, which merge to master.

```
brave@Thiru?? MINGW64 /s/GitDemo (master)
$ git branch -d GitWork
Deleted branch GitWork (was 2e374af).
```

18. Observe the log by executing “git log --oneline --graph --decorate”

```
brave@Thiru?? MINGW64 /s/GitDemo (master)
$ git log --oneline --graph --decorate
* 64cafd8 (HEAD -> master) Ignore merge backup files
*   f2ba564 Merged branch 'GitWork'
| \
| * 2e374af Updated hello.xml in GitWork branch
| * 0ec91e4 Added hello.xml in GitWork branch
* | 2e986c4 Added hello.xml in master branch
| /
*   bdfad9f Merge branch 'GitNewBranch'
| \
| * 949a89e Added newfile.txt in GitNewBranch
* | ec498f6 Add newfile.txt to master
| /
* 32807bf (origin/master) Added .gitignore to ignore .log files & log folders
* afe80b1 Added welcome.txt file
```