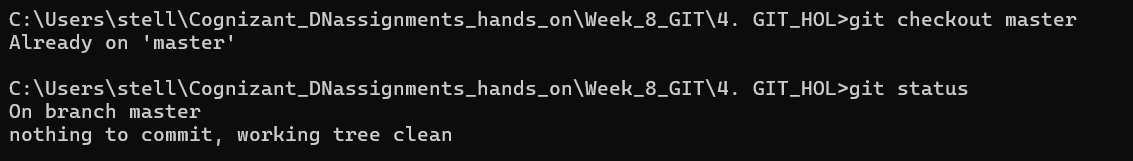
**4. GIT\_HOL**

**1. Verify if master is clean**



You should see nothing to commit, working tree clean.

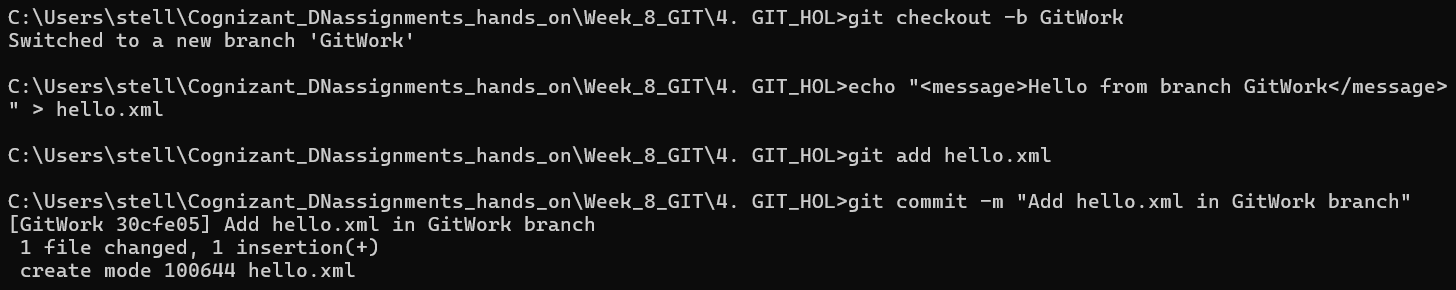
**2. Create branch GitWork and add hello.xml**

git checkout -b GitWork

echo "<message>Hello from branch GitWork</message>" > hello.xml

git add hello.xml

git commit -m "Add hello.xml in GitWork branch"



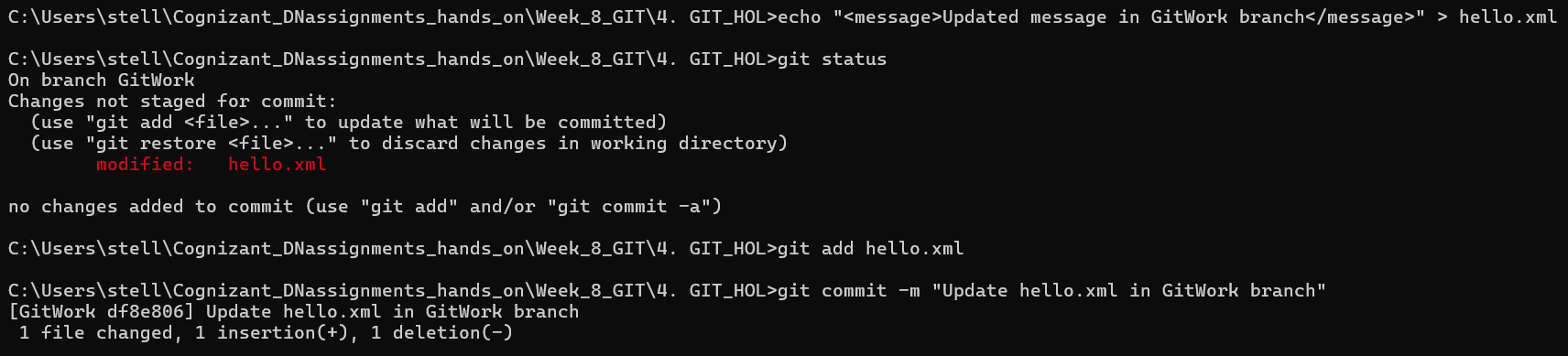
**3. Update hello.xml in branch**

echo "<message>Updated message in GitWork branch</message>" > hello.xml

git status

git add hello.xml

git commit -m "Update hello.xml in GitWork branch"



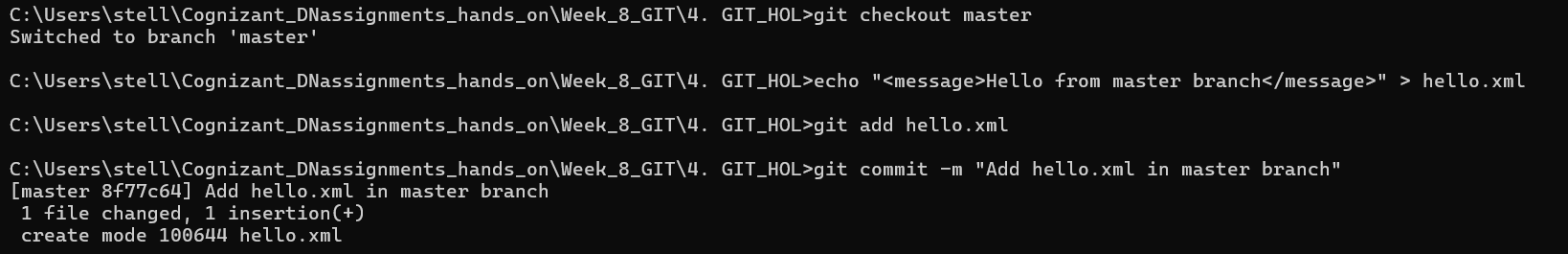
**4. Switch to master and create a different hello.xml**

git checkout master

echo "<message>Hello from master branch</message>" > hello.xml

git add hello.xml

git commit -m "Add hello.xml in master branch"



**5. Observe logs**

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



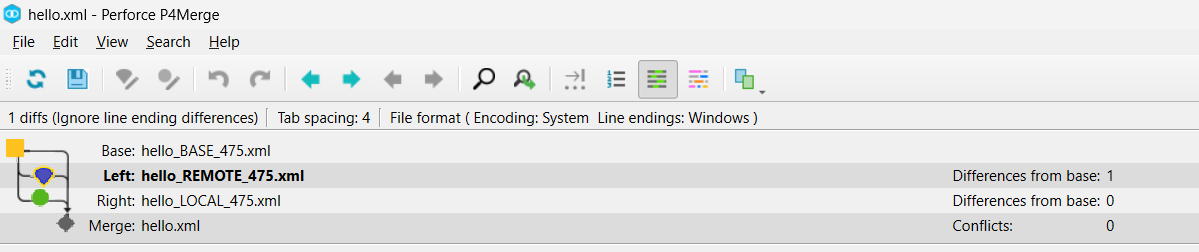
**6. Check differences**

git diff GitWork



**7. Visual diff with P4Merge**

git mergetool



**8. Merge GitWork into master**

git merge GitWork

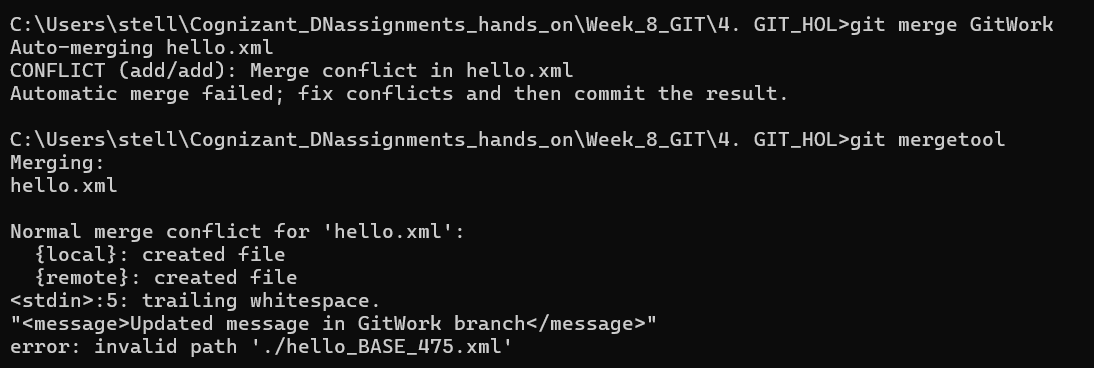
You should get a **merge conflict**.

**9. Resolve the conflict (3-way merge)**

* Open P4Merge automatically by:

git mergetool

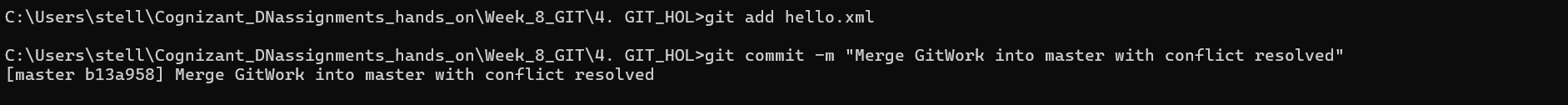
* Keep the correct changes manually, save, and close.



**10. Mark conflict as resolved and commit**

git add hello.xml

git commit -m "Merge GitWork into master with conflict resolved"

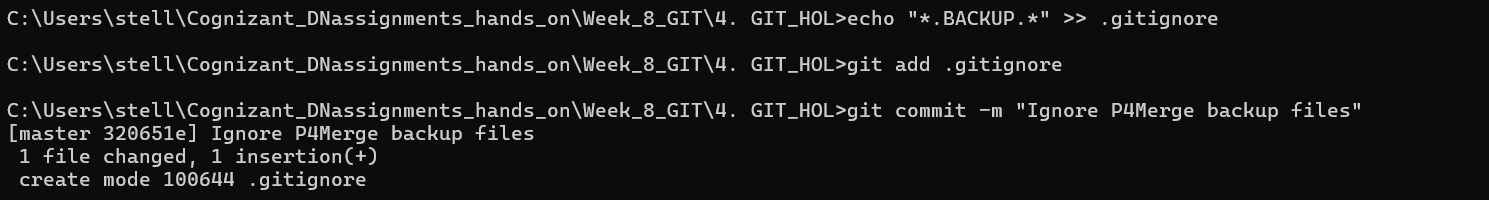


**11. Ignore backup files from P4Merge**

echo "\*.BACKUP.\*" >> .gitignore

git add .gitignore

git commit -m "Ignore P4Merge backup files"



**12. Delete branch**

git branch -d GitWork

**13. Final log**

git log --oneline --graph –decorate

