**WEEK 8 -GIT**

**TASK 4 :**

In this hands-on lab, you will learn how to:

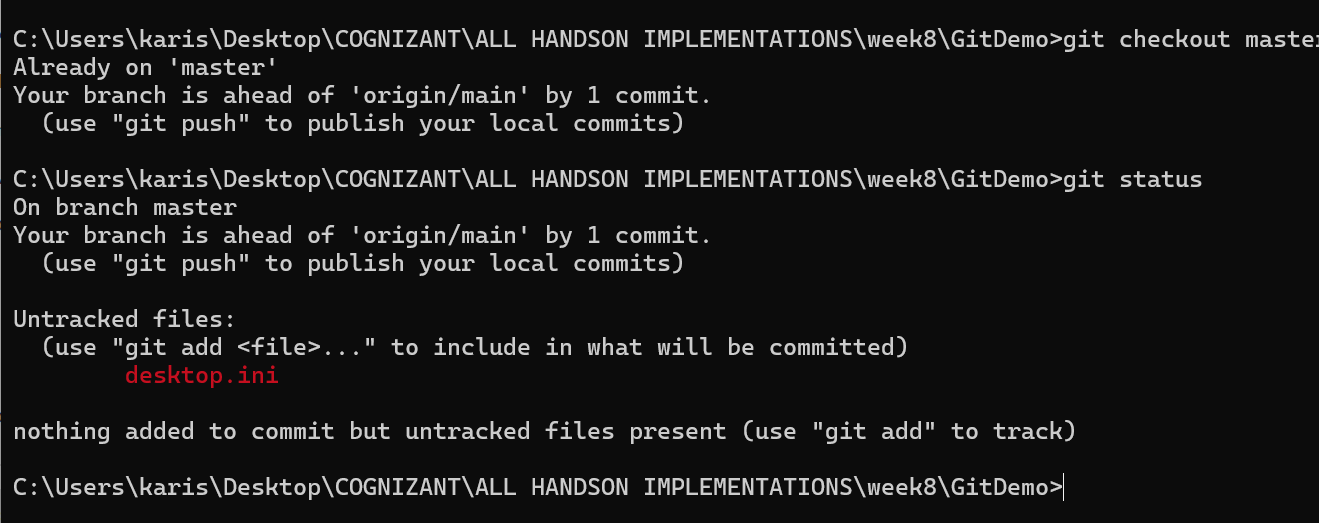
* Implement conflict resolution when multiple users are updating the trunk (or master) in such a way that it results into a conflict with the branch’s modification.

TASK AND OUTPUTS :

1. Verify if master is in clean state.

git checkout master

git status

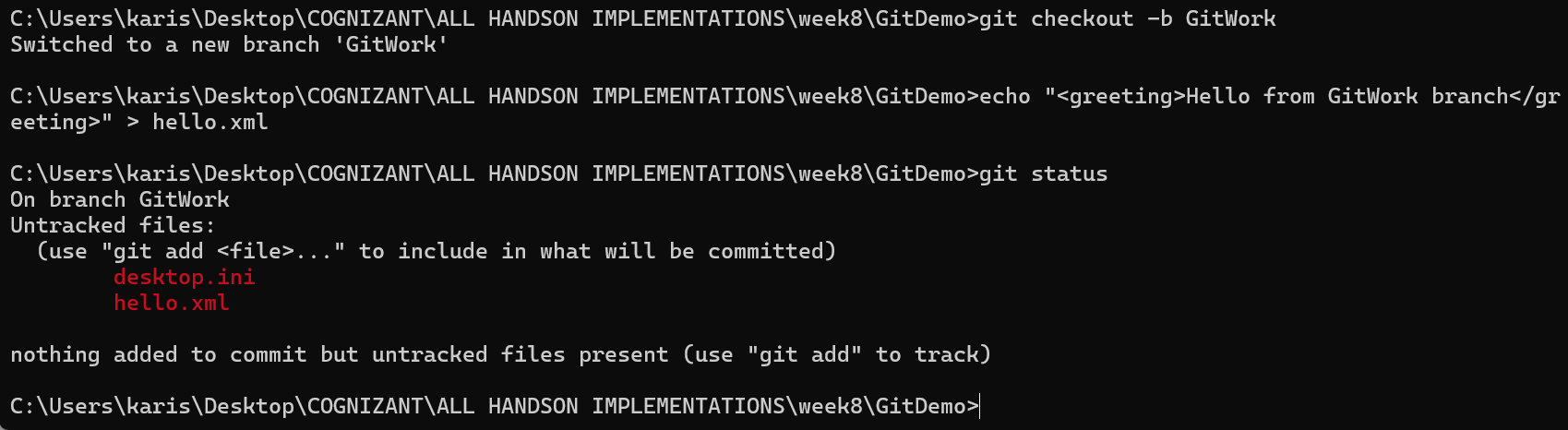


1. Create a branch “GitWork”. Add a file “hello.xml”.

git checkout -b GitWork

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

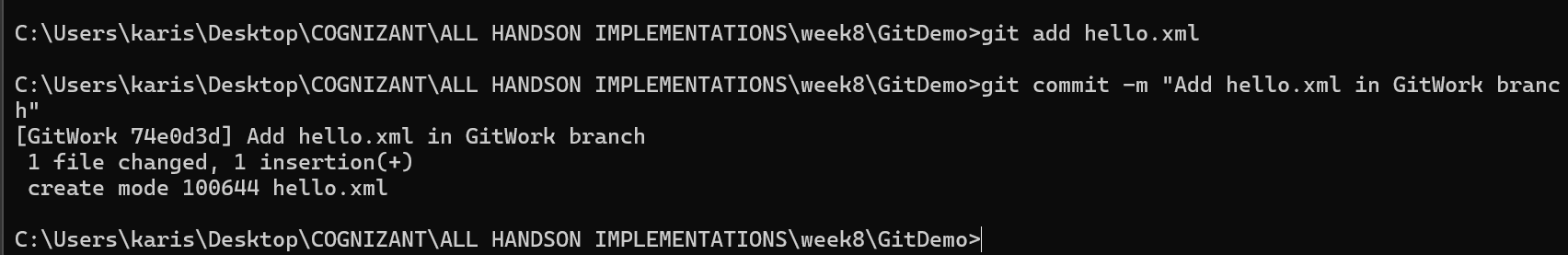
git status



1. Update the content of “hello.xml” and observe the status
2. Commit the changes to reflect in the branch

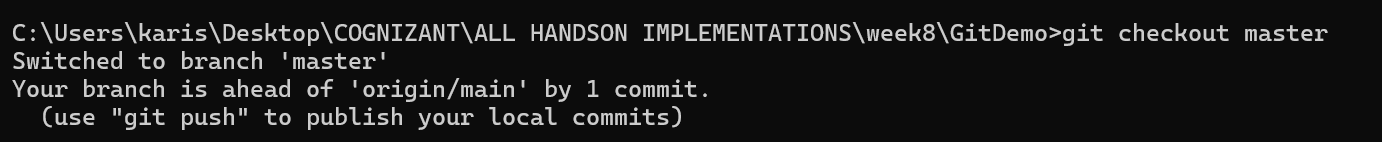
git add hello.xml

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



1. Switch to master.

git checkout master

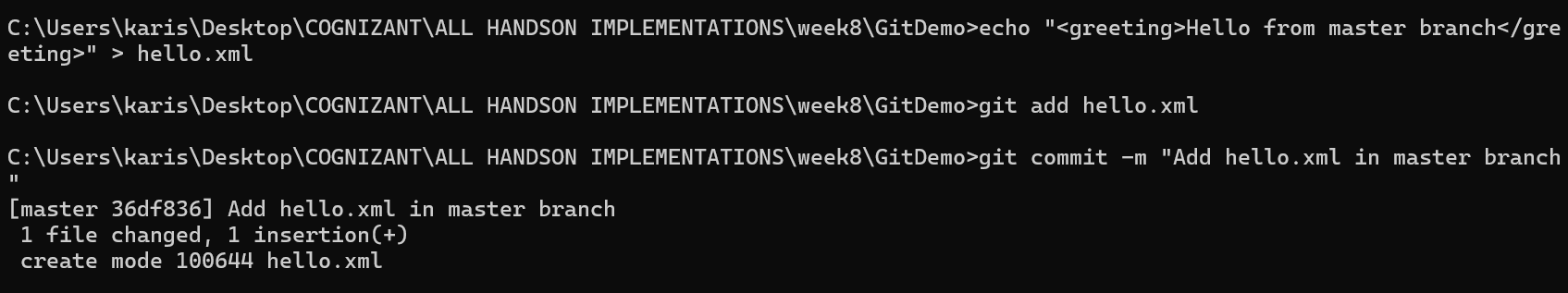


1. Add a file “hello.xml” to the master and add some different content than previous.

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

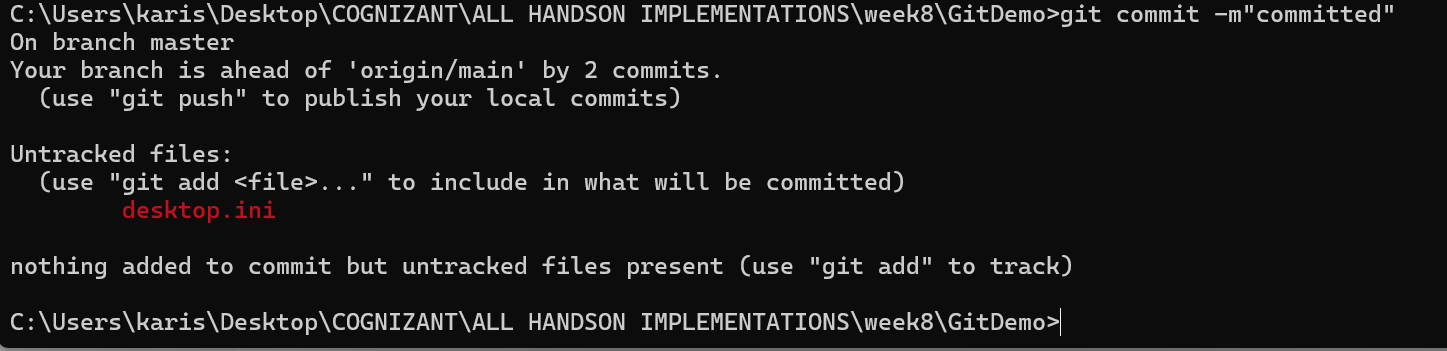
git add hello.xml

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



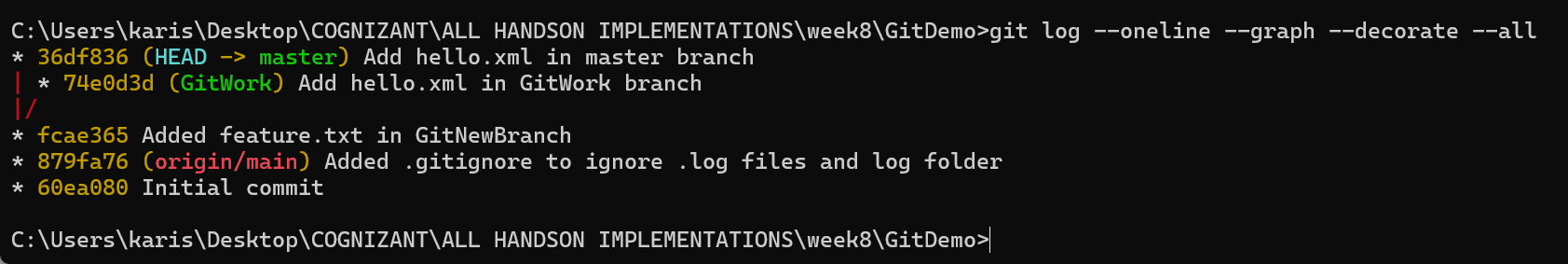
1. Commit the changes to the master

Git commit -m”committed”



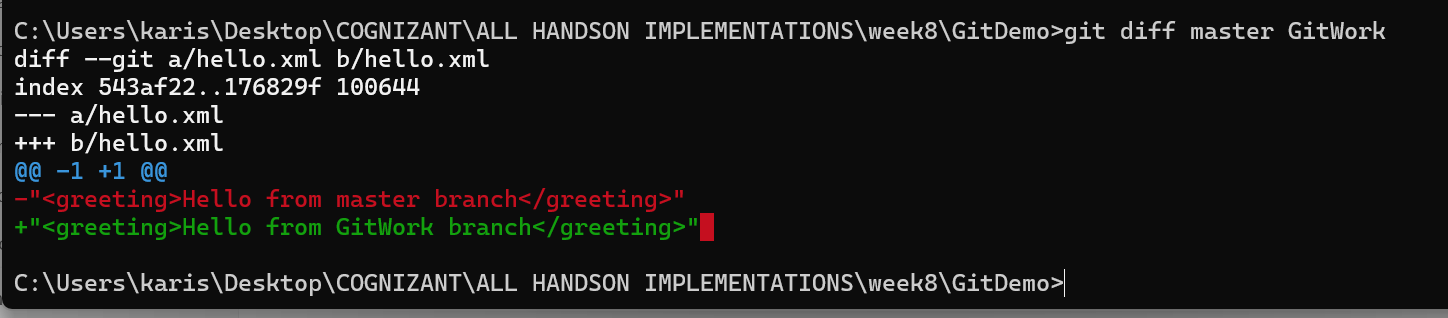
1. Observe the log by executing “git log –oneline –graph –decorate –all”

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



1. Check the differences with Git diff tool

git diff master GitWork

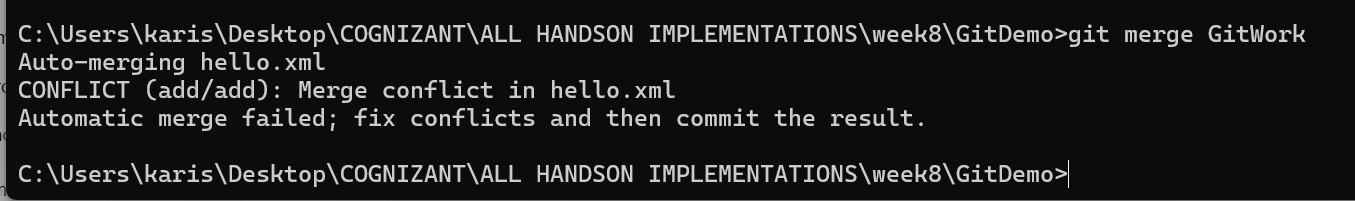


1. For better visualization, use P4Merge tool to list out all the differences between master and branch

git diff master GitWork

1. Merge the bran to the master

git merge GitWork

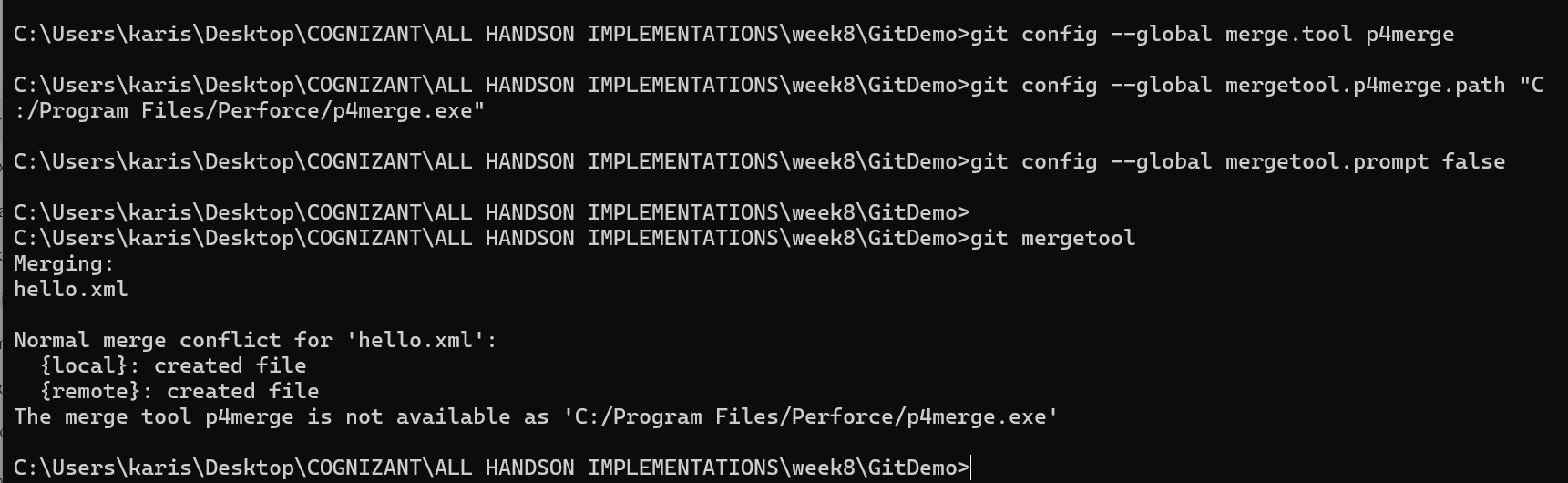


1. Observe the git mark up.
2. Use 3-way merge tool to resolve the conflict

git config --global merge.tool p4merge

git config --global mergetool.p4merge.path "C:/Program Files/Perforce/p4merge.exe"

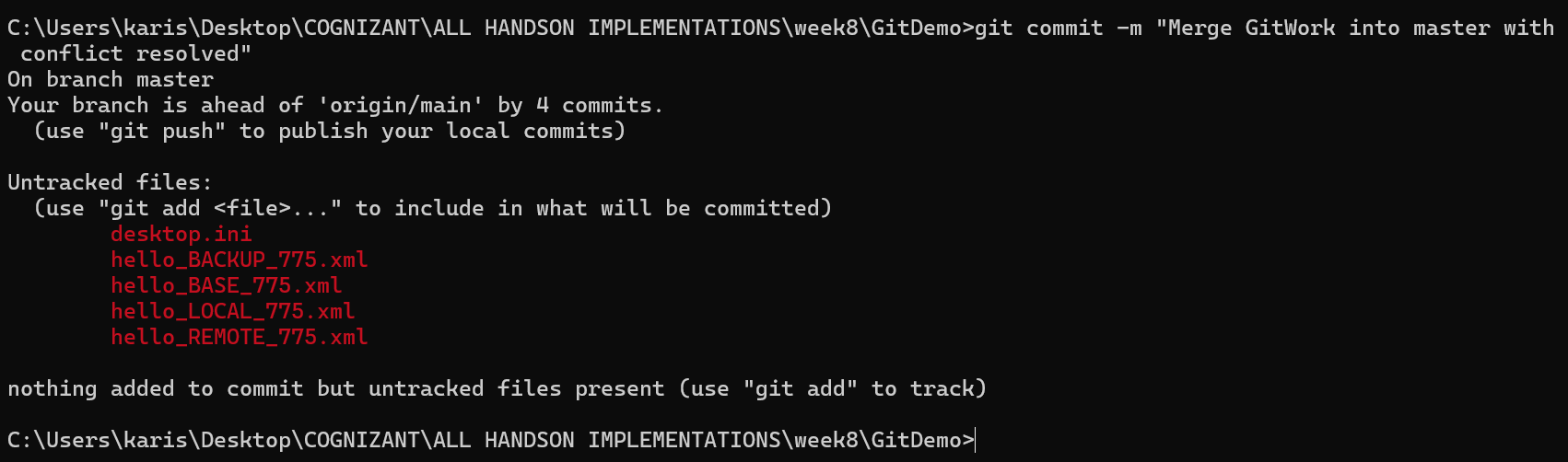
git config --global mergetool.prompt false



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

git add hello.xml

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



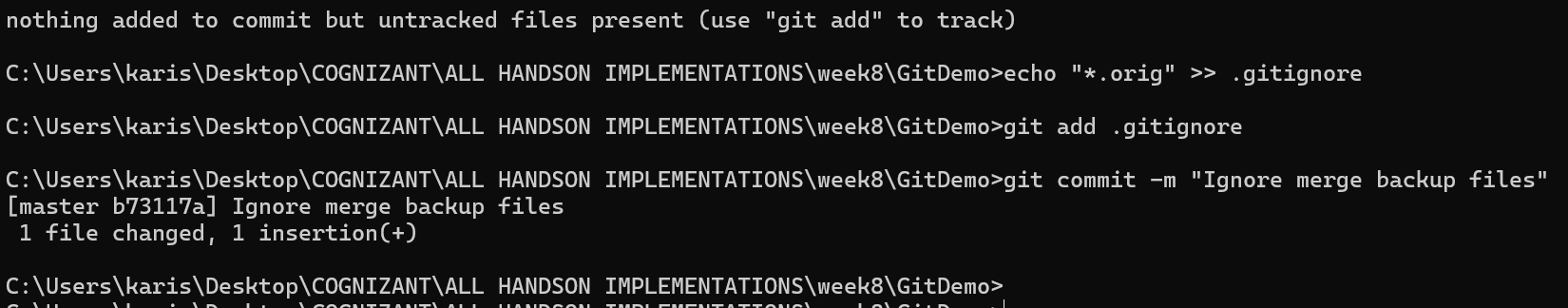
1. Observe the git status and add backup file to the .gitignore file.

echo "\*.orig" >> .gitignore

git add .gitignore

git commit -m "Ignore merge backup files"

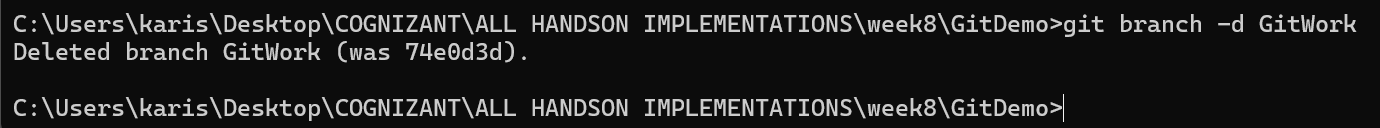
16. Commit the changes to the .gitignore



17. List out all the available branches

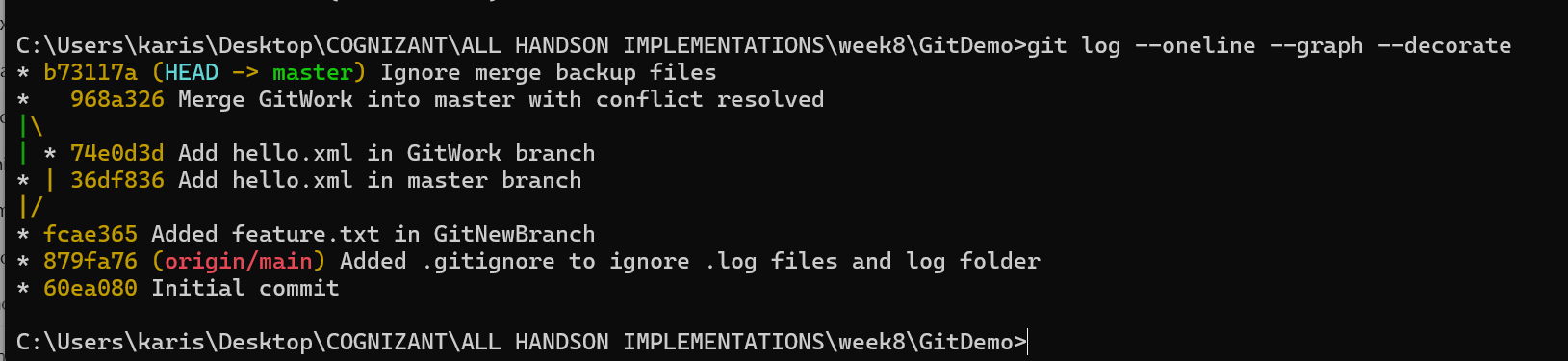
18. Delete the branch, which merge to master.

git branch -d GitWork



19. Observe the log by executing “git log –oneline –graph –decorate”

git log --oneline --graph –decorate



**DONE !!!!**