Name: Amogh Nagarkar

Super Set ID: 6403503

**Objectives**

* Explain how to resolve the conflict during merge.

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.

**Solution:**

cd ~/GitDemo

git status

git branch GitWork

git checkout GitWork

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

git status

git add hello.xml

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

git checkout master

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

git add hello.xml

git commit -m "Add hello.xml in master with different content"

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

git diff master GitWork

git merge GitWork

->open hello.xml

->edit to resolve manually

git add hello.xml

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

echo "\*~" >> .gitignore

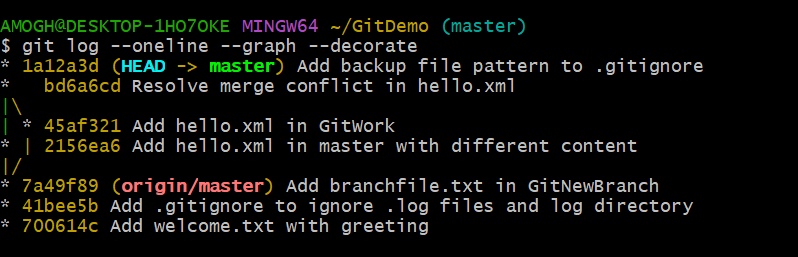
git add .gitignore

git commit -m "Add backup file pattern to .gitignore"

git branch

git branch -d GitWork

git log --oneline --graph –decorate



* Verified that the master branch was in a clean state.
* Created a new branch GitWork and added a file hello.xml with branch-specific content.
* Switched back to master, created another hello.xml file with different content, and committed it.
* Viewed the commit history and differences between master and GitWork.
* Attempted to merge GitWork into master, resulting in a merge conflict in hello.xml.
* Manually resolved the conflict by editing the file, then staged and committed the resolution.
* Added a backup file pattern to .gitignore and committed it.
* Deleted the merged branch GitWork and verified the final commit history.