# Explain how to resolve the conflict during merge.

## Theoretical Explanation

A merge conflict happens when Git tries to combine changes from two branches, but the same part of a file has been changed in different ways. Git cannot decide automatically which version to use, so it marks the file as conflicted and asks you to resolve it.  
  
When a conflict occurs, Git adds conflict markers inside the file:  
<<<<<<< HEAD  
Your changes (current branch)  
=======  
Their changes (branch you are merging)  
>>>>>>> BranchName  
  
To resolve a conflict:  
1. Identify conflicted files using 'git status'.  
2. Open the conflicted file in an editor or merge tool.  
3. Decide how to combine changes: keep your version, keep their version, keep both, or modify.  
4. Remove conflict markers and save the file.  
5. Stage the resolved file using 'git add <file>'.  
6. Commit the resolution using 'git commit -m "Resolved merge conflict"'.

## Brief & Easy Explanation

Think of it like two people editing the same sentence in a Google Doc at the same time. One writes: 'Hello from master branch!' and the other writes: 'Hello from GitWork branch!'. Git doesn't know which to keep, so it asks you to open the file and make the final decision. You look at both versions, choose the best or merge them together, save the file, and tell Git 'I’ve fixed it' by adding and committing.

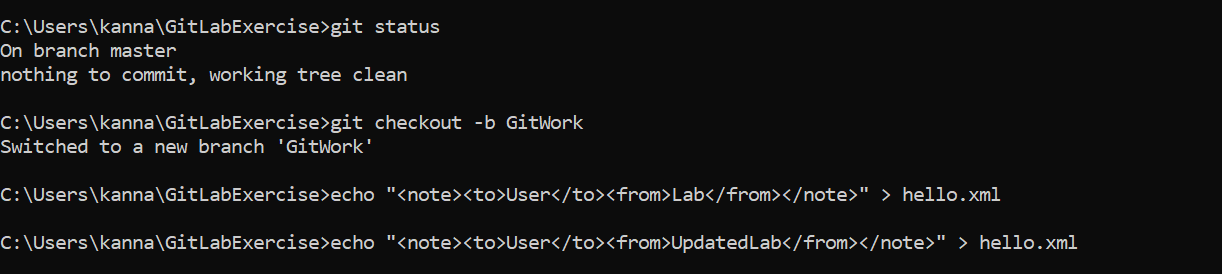
# Git Lab Commands

git status

git checkout -b GitWork

echo "<note><to>User</to><from>Lab</from></note>" > hello.xml

echo "<note><to>User</to><from>UpdatedLab</from></note>" > hello.xml

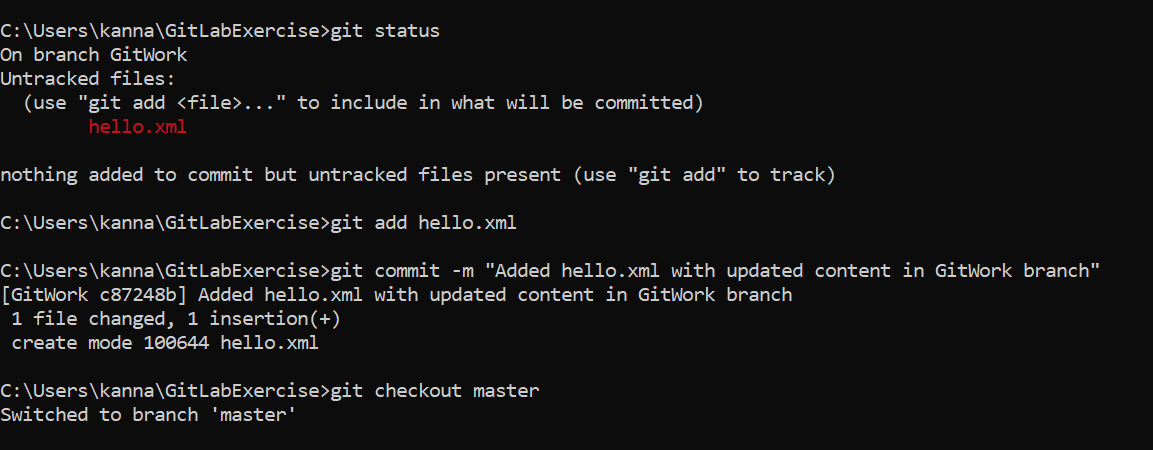


git status

git add hello.xml

git commit -m "Added hello.xml with updated content in GitWork branch"

git checkout master

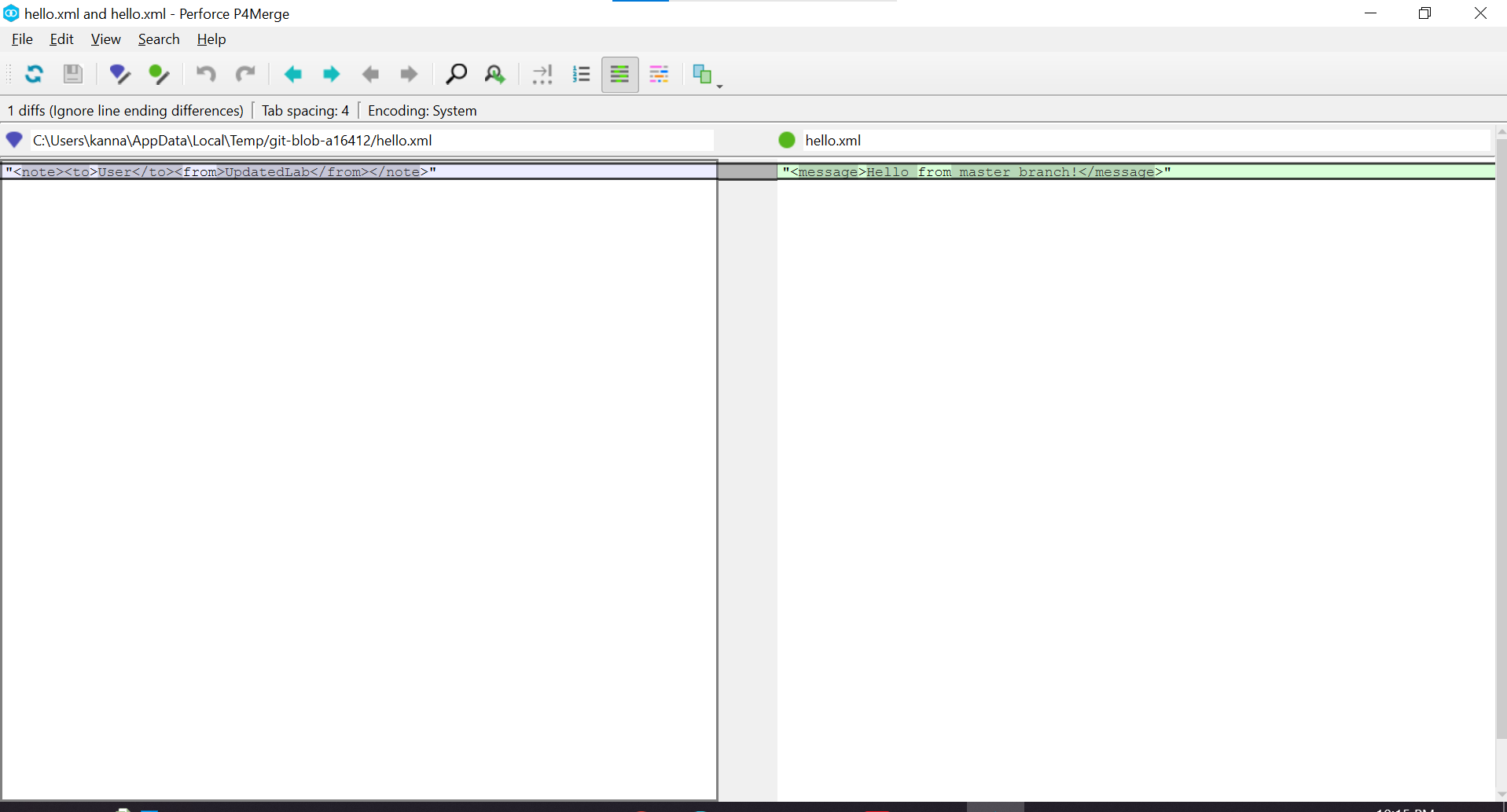
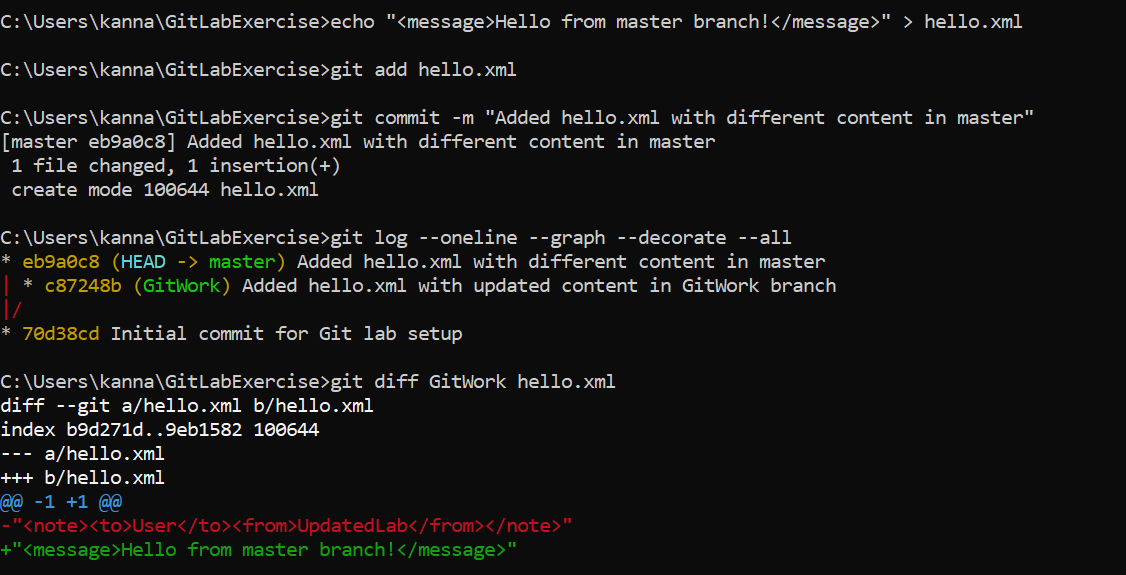


echo "<message>Hello from master branch!</message>" > hello.xml

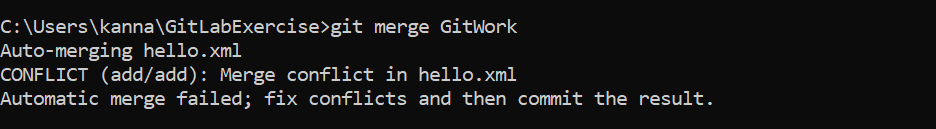
git add hello.xml  
git commit -m "Added hello.xml with different content in master"

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

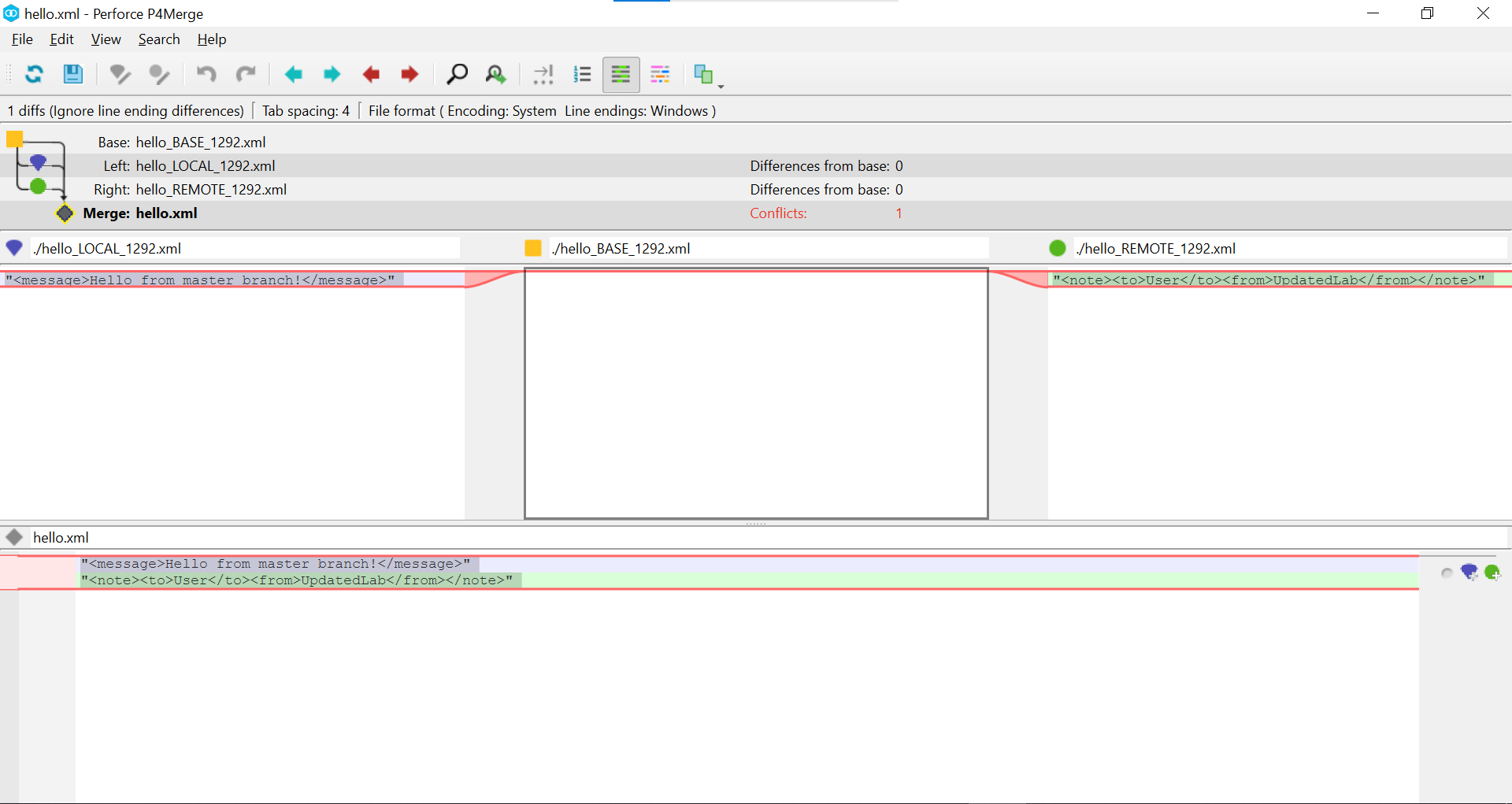
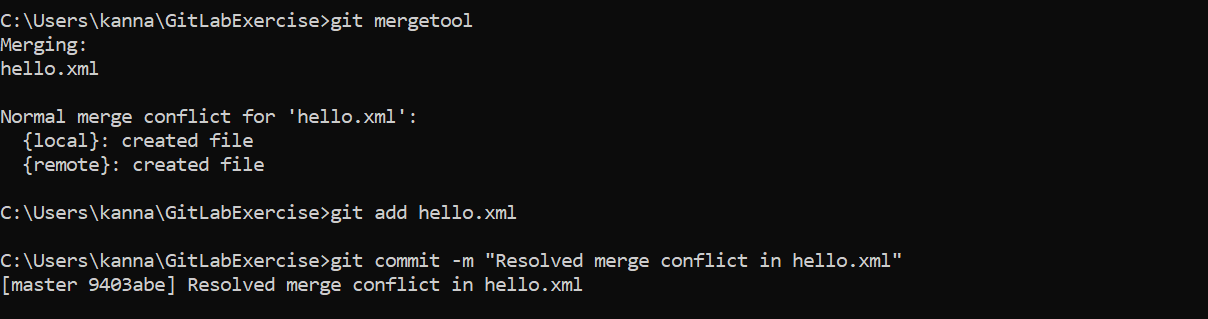
git diff GitWork hello.xml

git merge GitWork

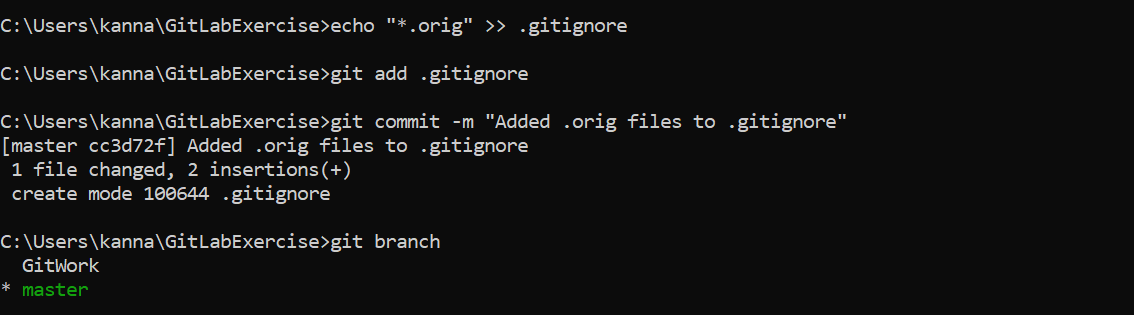
  
git mergetool  
git add hello.xml

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

echo "\*.orig" >> .gitignore  
git add .gitignore  
git commit -m "Added .orig files to .gitignore"

git branch



git branch -d GitWork  
git log --oneline --graph --decorate  
