**Objectives**

**Explain how to resolve the conflict during merge.**

When a merge conflict occurs, Git marks the conflicting sections in the file with  
<<<<<<<, =======, and >>>>>>>.  
To resolve:

1. Open the file with the conflict.
2. Decide which changes to keep — from current branch, other branch, or a combination.
3. Remove the conflict markers.
4. Save the file.
5. Stage it:

git add <filename>

1. Commit the merge:

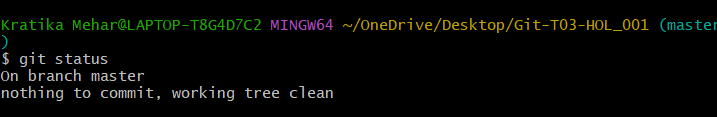
git commit

You can also use a merge tool (like P4Merge) to make this process easier visually.

**Step 1: Verify if master is in a clean state**

**Command:**

git status



**Step 2: Create a branch GitWork and add a file hello.xml**

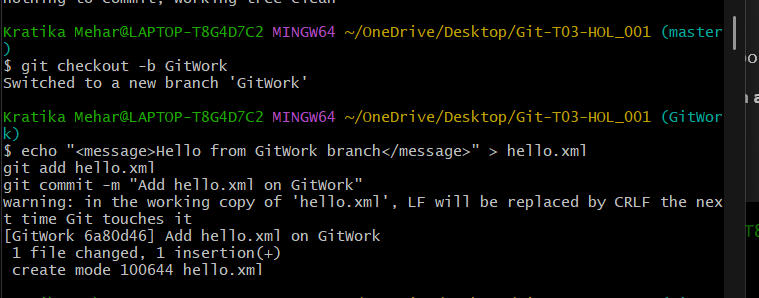
**Command:**

git checkout -b GitWork

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

git add hello.xml

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

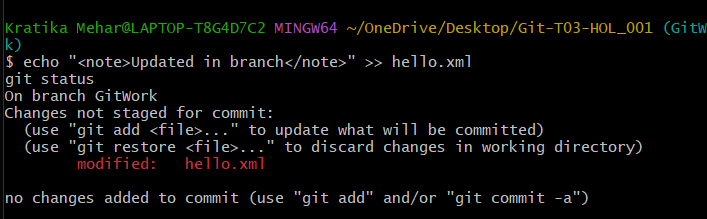


**Step 3: Update the content of hello.xml and observe status**

**Command:**

echo "<note>Updated in branch</note>" >> hello.xml

git status

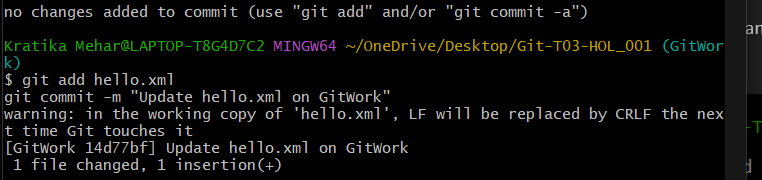


**Step 4: Commit the changes to the branch**

**Command:**

git add hello.xml

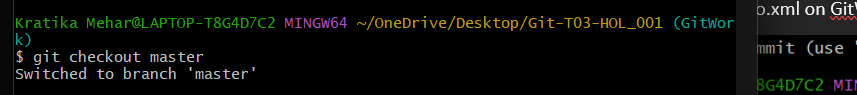
git commit -m "Update hello.xml on GitWork"



**Step 5: Switch to master**

**Command:**

git checkout master



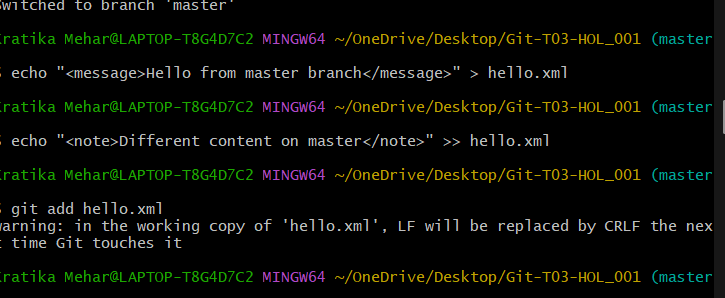
**Step 6: Add a file hello.xml to master with different content**

**Command:**

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

echo "<note>Different content on master</note>" >> hello.xml

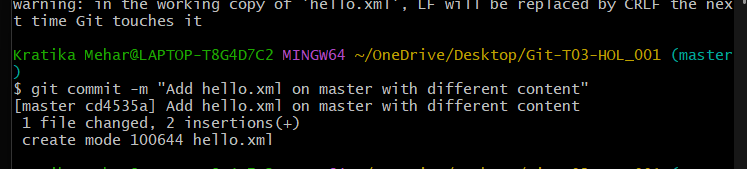
git add hello.xml



**Step 7: Commit the changes to master**

**Command:**

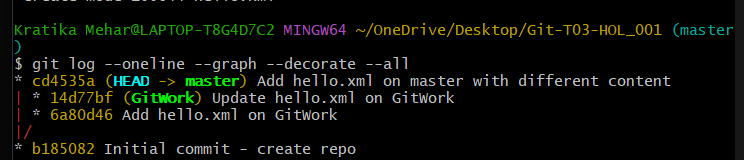
git commit -m "Add hello.xml on master with different content"



**Step 8: Observe the log**

**Command:**

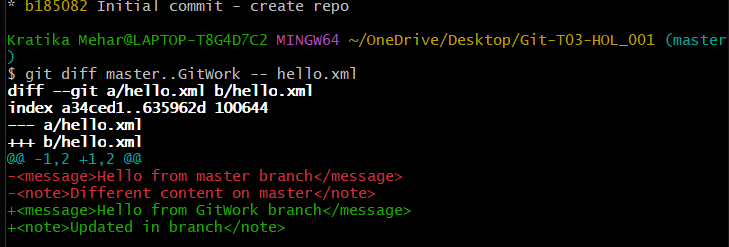
git log --oneline --graph --decorate –all



**Step 9: Check the differences using Git diff**

**Command:**

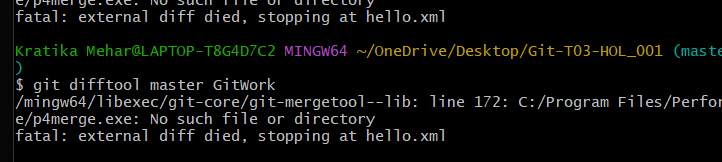
git diff master..GitWork -- hello.xml



**Step 10: Use P4Merge to visualize differences *(Optional)***

**Command (if P4Merge is set up):**

git difftool master GitWork -- hello.xml

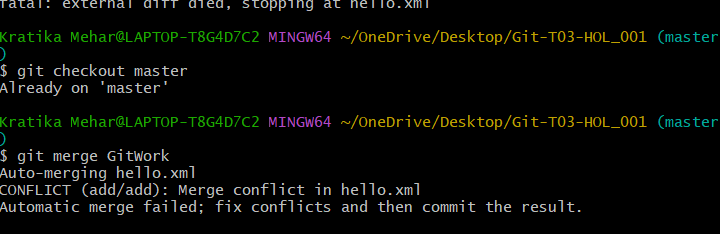


**Step 11: Merge the branch into master**

**Command:**

git checkout master

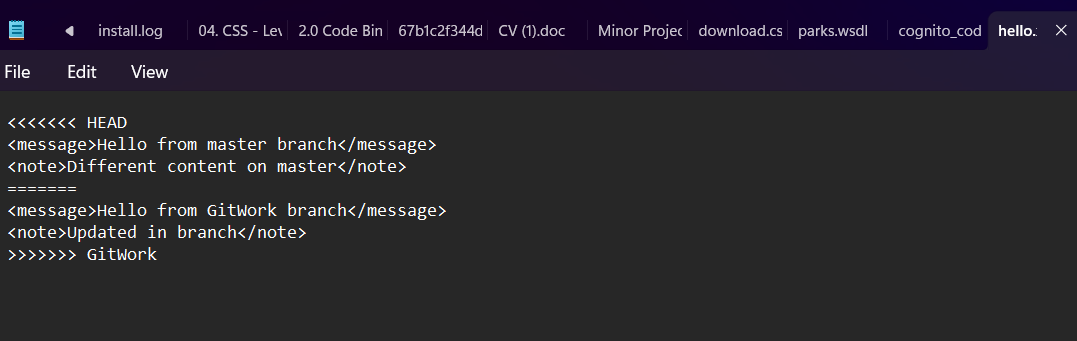
git merge GitWork



**Step 12: Observe the Git Markup**

**Command:**

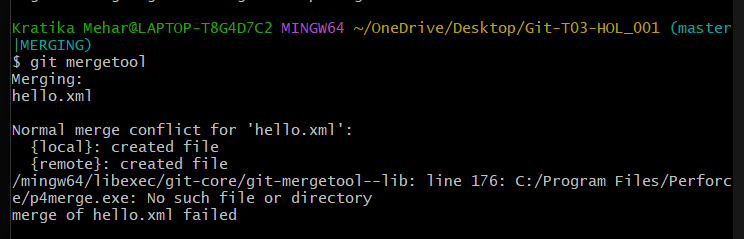
notepad hello.xml



**Step 13: Resolve the conflict using 3-way merge tool (P4Merge)**

**Command:**

git mergetool

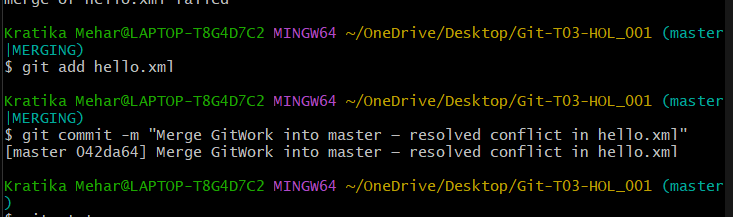


**Step 14: Commit the resolved changes**

**Command:**

git add hello.xml

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

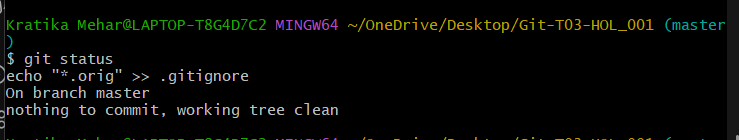


**Step 15: Observe status and add backup files to .gitignore**

**Command:**

git status

echo "\*.orig" >> .gitignore

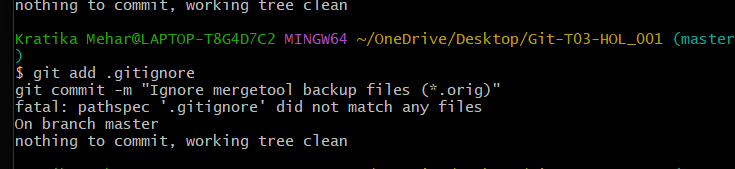


**Step 16: Commit .gitignore changes**

**Command:**

git add .gitignore

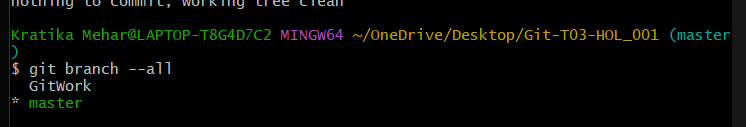
git commit -m "Ignore merge backup files"



**Step 17: List all available branches**

**Command:**

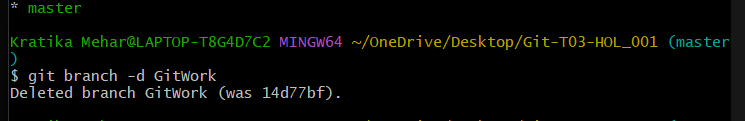
git branch –all



**Step 18: Delete the merged branch**

**Command:**

git branch -d GitWork



**Step 19: Observe the log again**

**Command:**

git log --oneline --graph –decorate

