**Step 1: Verify Master Branch is Clean**  
**Description:**  
Checked that the master branch has no uncommitted changes before starting the lab.  
**Commands Used:**

git checkout master

git status

**Step 2: Create and Switch to GitWork Branch**  
**Description:**  
Created a new branch GitWork from master and switched to it.  
**Commands Used:**

git branch GitWork

git checkout GitWork

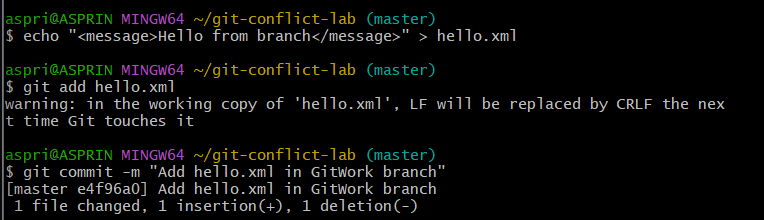
**Step 3: Add hello.xml in GitWork**  
**Description:**  
Created hello.xml with unique content for the GitWork branch and committed the change.  
**Commands Used:**

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

git add hello.xml

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

**Screenshot(s):**



**Step 4: Switch to Master and Add hello.xml**  
**Description:**  
Switched back to master and created a different version of hello.xml, committing the change.  
**Commands Used:**

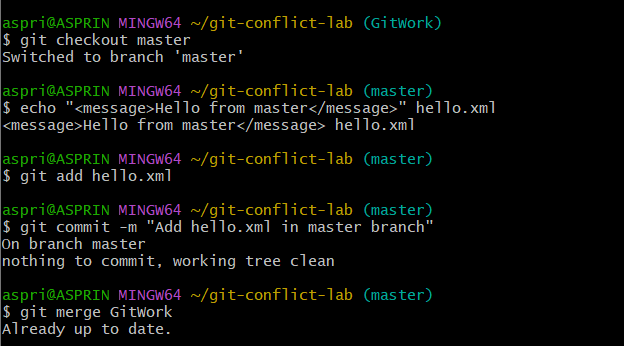
git checkout master

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

git add hello.xml

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

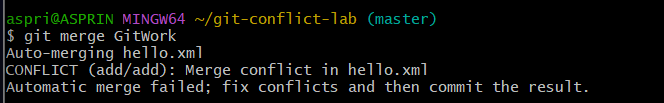
**Screenshot(s):**



**Step 5: Merge and Trigger Conflict**  
**Description:**  
Merged GitWork into master, resulting in a merge conflict in hello.xml.  
**Commands Used:**

git merge GitWork

**Screenshot(s):**

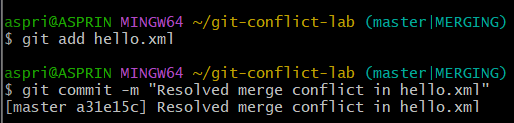


**Step 6: Resolve Merge Conflict**  
**Description:**  
Opened hello.xml, manually resolved the conflict by combining both changes, removed conflict markers, and committed the result.  
**Commands Used:**

git add hello.xml

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

**Screenshot(s):**



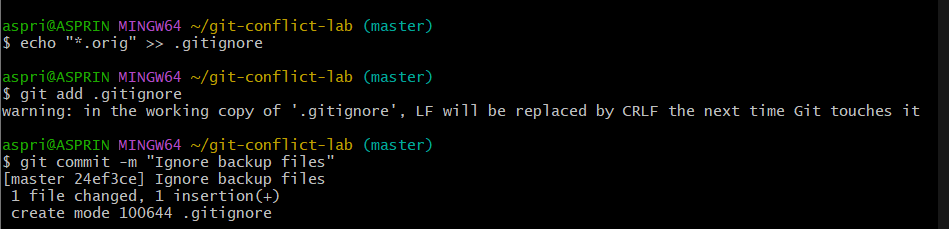
**Step 7: Update .gitignore**  
**Description:**  
Ignored backup files by adding a rule to .gitignore and committing the change.  
**Commands Used:**

echo "\*.orig" >> .gitignore

git add .gitignore

git commit -m "Ignore backup files"

**Screenshot(s):**



**Step 8: Delete GitWork Branch**  
**Description:**  
Deleted the merged GitWork branch.  
**Commands Used:**

git branch -d GitWork

**Screenshot(s):**



**Step 9: View Commit History**  
**Description:**  
Viewed the commit history in a graphical format to confirm the merge and branch deletion.  
**Commands Used:**

git log --oneline --graph --decorate

**Screenshot(s):**

