**Git Hands-on Lab Commands for Conflict Resolution**

The following commands are used to perform the tasks outlined in the lab assignment for resolving merge conflicts.

**Step 1: Set up the branches and conflicting files:**

1. **Verify the master branch is clean:** git status
2. **Create and switch to a new branch:** git branch GitWork git checkout GitWork
3. **Create and modify a file in the new branch:** echo "Content from GitWork branch" > hello.xml
4. **Commit the changes:** git add hello.xml git commit -m "Added hello.xml to GitWork branch"

**Step 2: Create a conflict on the master branch:**

1. **Switch back to master:** git checkout master
2. **Modify the same file:** echo "Different content from master" > hello.xml
3. **Commit the changes:** git add hello.xml git commit -m "Added hello.xml to master branch with different content"
4. **Observe the log:** git log --oneline --graph --decorate --all

**Step 3: Resolve the merge conflict:**

1. **Attempt to merge:** git merge GitWork
2. **Use a merge tool to resolve:** git mergetool
3. **Finalize the merge:** git add hello.xml git commit -m "Merged GitWork branch and resolved conflict"
4. **Clean up:** echo "\*.orig" >> .gitignore git add .gitignore git commit -m "Added merge tool backup file to .gitignore"

**Step 4: Clean up and verify:**

1. **List branches:** git branch -a
2. **Delete the merged branch:** git branch -d GitWork
3. **Observe the final log:** git log --oneline --graph --decorate