To practice the complete Git workflow including branching, file versioning, conflict resolution, using merge tools, and final cleanup using Git Bash commands.

**Step-by-Step Execution**

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

git status

**Output**: Working directory clean — No changes to commit.

**🔹 Step 2: Create a new branch GitWork and add a file**

git checkout -b GitWork

echo "<message>This is GitWork update</message>" > hello.xml

**🔹 Step 3: Update the content of hello.xml**

notepad hello.xml

Edited the file to contain:

<message>This is GitWork update</message>

**🔹 Step 4: Commit changes in the GitWork branch**

git add hello.xml

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

**🔹 Step 5: Switch back to the master branch**

git checkout master

**🔹 Step 6: Add a different version of hello.xml in master**

echo "<message>This is a master update</message>" > hello.xml

**🔹 Step 7: Commit the file to master**

git add hello.xml

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

**🔹 Step 8: View the log with graphical visualization**

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

**Output**: Shows both branches diverged with different commits on hello.xml.

**🔹 Step 9: View file differences using Git diff**

git diff GitWork

**Output**: Highlights the line difference between the two branches.

**🔹 Step 10: Use merge tool (vimdiff / P4Merge) for visualization**

P4Merge was not configured, so fallback was to:

git mergetool

**Tool used**: vimdiff (or manual merge using Notepad).

**🔹 Step 11: Attempt to merge the branch into master**

git merge GitWork

**Output**: Merge conflict detected in hello.xml.

**🔹 Step 12: Observe conflict markers**

Conflict markers appeared:

<<<<<<< HEAD

<message>This is a master update</message>

=======

<message>This is GitWork update</message>

>>>>>>> GitWork

**🔹 Step 13: Use 3-way merge tool / Manual edit**

Opened file in Notepad:

notepad hello.xml

Resolved by keeping both messages:

<message>This is a master update</message>

<message>This is GitWork update</message>

Saved and exited.

**🔹 Step 14: Finalize the conflict resolution and commit**

git add hello.xml

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

**🔹 Step 15: Observe .orig backup file and update .gitignore**

git status

echo "\*.orig" >> .gitignore

**🔹 Step 16: Commit .gitignore update**

git add .gitignore

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

**🔹 Step 17: List available branches**

git branch

**Output**:

* master
* GitWork

**🔹 Step 18: Delete the merged branch**

git branch -d GitWork

**🔹 Step 19: Final check of commit history**

git log --oneline --graph --decorate

**Output**: Shows merged history of both branches.

**Key Commands Used**

git status

git checkout -b GitWork

git add hello.xml

git commit -m "message"

git checkout master

git merge GitWork

git mergetool

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

git diff

git branch -d GitWork