# 1. Verify if master is in clean state

git checkout master

git status

# 2. Create a branch “GitWork” and switch to it

git checkout -b GitWork

# Add a file “hello.xml”

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

git add hello.xml

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

# 3. Update the content of “hello.xml” and observe the status

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

git status

# 4. Commit the changes to reflect in the branch

git add hello.xml

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

# 5. Switch to master

git checkout master

# 6. Add a file “hello.xml” to master with different content

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

# 7. Commit the changes to master

git add hello.xml

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

# 8. Observe the log with graph

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

# 9. Check the differences between master and GitWork (CLI diff)

git diff master GitWork

# 10. Use P4Merge to list differences (requires P4Merge installed & configured)

git config --global diff.tool p4merge

git config --global difftool.p4merge.path "C:/Program Files/Perforce/p4merge.exe"

git difftool master GitWork

# 11. Merge the branch into master

git merge GitWork

# 12. Observe git markup (conflicts will be shown here)

# (No command — just check terminal output after merge)

# 13. Use 3-way merge tool (P4Merge) to resolve conflicts

git mergetool

# 14. Once conflicts are resolved, commit the changes to master

git commit -m "Merge GitWork into master with conflict resolution"

# 15. Observe git status and add backup files to .gitignore

git status

echo "\*~" >> .gitignore # Example: ignoring backup files ending with ~

git add .gitignore

# 16. Commit the changes to .gitignore

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

# 17. List out all available branches

git branch -a

# 18. Delete the branch which was merged to master

git branch -d GitWork

# 19. Observe the log after merge

git log --oneline --graph --decorate