# 4. Git HOL – Branching, Merging & Conflict Resolution

## 1️⃣ Verify master is in clean state

git checkout master  
git status  
If you see uncommitted changes, commit or stash them before proceeding.  
A screen shot of a computer program

AI-generated content may be incorrect.

## 2️⃣ Create branch “GitWork” and add file

git branch GitWork  
git checkout GitWork  
echo "<note>Hello from GitWork branch</note>" > hello.xml  
git add hello.xml  
git commit -m "Added hello.xml in GitWork branch"  
A screen shot of a computer

AI-generated content may be incorrect.

## 3️⃣ Update content in branch

echo "<note>Updated content in GitWork branch</note>" > hello.xml  
git status  
git add hello.xml  
git commit -m "Updated hello.xml in GitWork branch"  
A screenshot of a computer screen

AI-generated content may be incorrect.

## 4️⃣ Switch to master and add conflicting file

git checkout master  
echo "<note>Master branch version</note>" > hello.xml  
git add hello.xml  
git commit -m "Added conflicting hello.xml in master"  
A screenshot of a computer program

AI-generated content may be incorrect.

## 5️⃣ Observe commit history

git log --oneline --graph --decorate --all  
A screen shot of a computer screen

AI-generated content may be incorrect.

## 6️⃣ Compare differences

git diff GitWork  
git mergetool

## 7️⃣ Merge branch into master (will cause conflict)

git merge GitWork  
# Git will indicate a conflict in hello.xml

A black background with white text

AI-generated content may be incorrect.

## 8️⃣ Resolve conflict using 3-way merge tool

# Open hello.xml and choose correct content manually or:  
git mergetool  
# After resolving:  
git add hello.xml  
git commit -m "Resolved merge conflict in hello.xml"

## 9️⃣ Update .gitignore for backup files

echo "\*.orig" >> .gitignore  
git add .gitignore  
git commit -m "Updated .gitignore to exclude backup files"

## 🔟 Delete merged branch

git branch -d GitWork  
git log --oneline --graph --decorate