# Git Branch Creation, Merge with Conflict Resolution — CLI Commands & Outputs

## 1) Verify if master is in clean state

$ git status  
On branch master  
nothing to commit, working tree clean

## 2) Create a branch “GitWork” and add hello.xml

$ git checkout -b GitWork  
Switched to a new branch 'GitWork'  
$ echo '<message>Hello from GitWork</message>' > hello.xml

## 3) Update hello.xml and observe the status

$ echo '<note>Updated in GitWork branch</note>' >> hello.xml  
$ git status  
On branch GitWork  
Changes not staged for commit:  
 modified: hello.xml

## 4) Commit the changes to reflect in the branch

$ git add hello.xml  
$ git commit -m 'Updated hello.xml in GitWork'  
[GitWork abc1234] Updated hello.xml in GitWork  
 1 file changed, 2 insertions(+)

## 5) Switch to master

$ git checkout master  
Switched to branch 'master'

## 6) Add hello.xml to master with different content

$ echo '<greeting>Hello from master branch</greeting>' > hello.xml  
$ git add hello.xml  
$ git commit -m 'Added hello.xml in master'  
[master def5678] Added hello.xml in master  
 1 file changed, 1 insertion(+)  
 create mode 100644 hello.xml

## 7) Observe log with all branches

$ git log --oneline --graph --decorate --all  
\* def5678 (HEAD -> master) Added hello.xml in master  
| \* abc1234 (GitWork) Updated hello.xml in GitWork  
|/   
\* 1234567 Initial commit

## 8) Check differences between master and branch

$ git diff master GitWork  
diff --git a/hello.xml b/hello.xml  
index 1111111..2222222 100644  
--- a/hello.xml  
+++ b/hello.xml  
@@  
-<greeting>Hello from master branch</greeting>  
+<message>Hello from GitWork</message>  
+<note>Updated in GitWork branch</note>

## 9) Use P4Merge tool to view differences

$ git mergetool --tool=p4merge  
Merging:  
hello.xml  
Normal merge conflict for 'hello.xml':  
 {local}: modified file  
 {remote}: modified file  
Hit return to start merge resolution tool (p4merge)...

## 10) Merge GitWork into master (conflict expected)

$ git merge GitWork  
Auto-merging hello.xml  
CONFLICT (content): Merge conflict in hello.xml  
Automatic merge failed; fix conflicts and then commit the result.

## 11) Use 3-way merge tool to resolve conflict

$ git mergetool  
Merging:  
hello.xml  
Normal merge conflict for 'hello.xml':  
Hit return to start merge resolution tool...

## 12) Commit the changes after conflict resolution

$ git add hello.xml  
$ git commit -m 'Merge GitWork into master with conflict resolved'  
[master 789abcd] Merge GitWork into master with conflict resolved

## 13) Add backup file to .gitignore

$ echo '\*.orig' >> .gitignore  
$ git add .gitignore  
$ git commit -m 'Added .gitignore for merge backups'  
[master efg4567] Added .gitignore for merge backups

## 14) List all available branches

$ git branch  
\* master  
 GitWork

## 15) Delete the merged branch

$ git branch -d GitWork  
Deleted branch GitWork (was abc1234).

## 16) Observe final log

$ git log --oneline --graph --decorate  
\* efg4567 (HEAD -> master) Added .gitignore for merge backups  
\* 789abcd Merge GitWork into master with conflict resolved  
|\   
| \* abc1234 Updated hello.xml in GitWork  
\* | def5678 Added hello.xml in master  
|/   
\* 1234567 Initial commit