**Git Merge Conflict Resolution – Hands-on Lab**

**Objectives**

* Explain how to resolve the conflict during merge.

**In this hands-on lab, you will learn how to:**

* Implement conflict resolution when multiple users are updating the trunk (or master) in such a way that it results in a conflict with the branch’s modification.

**Prerequisites**

* Hands-on ID: **Git-T03-HOL\_001**

**Notes**

Please follow the below steps for creating a free account in GitHub.  
Do not use cognizant credentials to login to GitHub.

Estimated time to complete this lab: **30 minutes**  
Each instruction expects a **Git Bash command** to be executed.

**Steps**

1. **Verify if master is in clean state**
2. git status
3. **Create a branch “GitWork” and add a file “hello.xml”**
4. git branch GitWork
5. git checkout GitWork
6. echo "<message>Hello from GitWork branch</message>" > hello.xml
7. **Update the content of hello.xml and observe the status**
8. echo "<message>Updated content in GitWork branch</message>" > hello.xml
9. git status
10. **Commit the changes to reflect in the branch**
11. git add hello.xml
12. git commit -m "Added hello.xml in GitWork branch"
13. **Switch to master**
14. git checkout master
15. **Add a file hello.xml to the master with different content**
16. echo "<message>Hello from master branch</message>" > hello.xml
17. **Commit the changes to the master**
18. git add hello.xml
19. git commit -m "Added hello.xml in master branch"
20. **Observe the log**
21. git log --oneline --graph --decorate --all
22. **Check the differences using Git diff tool**
23. git diff GitWork
24. **For better visualization, use P4Merge tool**
25. git mergetool
26. **Merge the branch into the master**
27. git merge GitWork
28. **Observe the Git markup**
    * The file will have conflict markers like:
    * <<<<<<< HEAD
    * Content from master
    * =======
    * Content from GitWork
    * >>>>>>> GitWork
29. **Use 3-way merge tool to resolve the conflict**
30. git mergetool
31. **Commit the changes after conflict resolution**
32. git add hello.xml
33. git commit -m "Resolved merge conflict in hello.xml"
34. **Observe the Git status and add backup file to .gitignore**
35. git status
36. echo "\*.orig" >> .gitignore
37. **Commit the .gitignore changes**
38. git add .gitignore
39. git commit -m "Added .orig files to .gitignore"
40. **List out all available branches**
41. git branch -a
42. **Delete the merged branch**
43. git branch -d GitWork
44. **Observe the log again**
45. git log --oneline --graph --decorate

