**GIT HANDSON**

**Exercise – 4**

A **Merge conflict** occurs when two branches change the same part of a file and Git can’t decide which version to keep.

1.Identify the Conflict

First, Git will stop the merge process and tell you which file (or files) has a conflict. You can also use the git status command to see a list of "unmerged paths," which are the files with conflicts.

2.Manually Edit the File

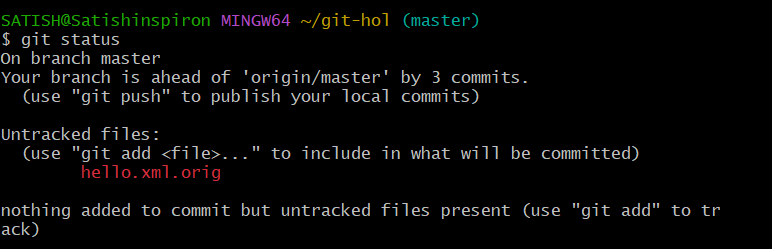
Edit the file to fix the conflict. need to decide what the final version of the code should look like.

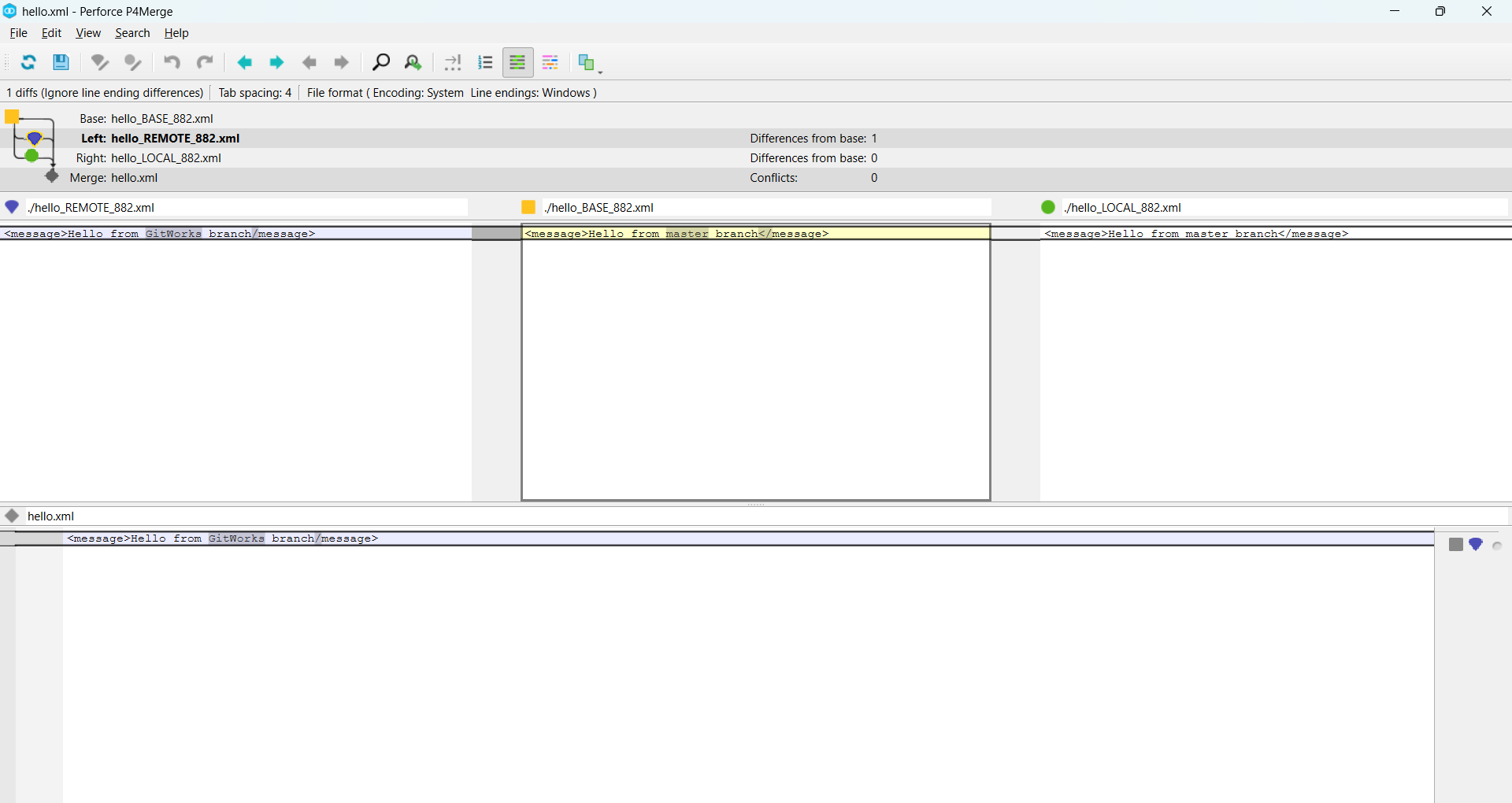
* Choose one version: might decide to keep only your changes and delete the other version.
* Keep the other version: might decide the other branch has the correct code and delete yours.
* Combine both: can manually edit the code to incorporate parts of both changes.

3. Commit the Fix

Once fixed all the conflicts in the file (or files), need to tell Git that the conflict is resolved.

* Stage the fixed file: Use the git add <filename> command for each file fixed. This tells Git that have resolved the conflict in that file.
* Commit the merge: Once all conflicted files are staged, run git commit. Git will open a pre-populated commit message (e.g., "Merge branch 'new-feature'"). can just save this message to complete the merge.



****

