**Exercise 4**

Merge conflicts happen when Git is unable to automatically merge changes from two different branches. This typically occurs when the same line of code has been edited in both branches.1 Resolving a conflict involves manually deciding which changes to keep.2

### Resolving a Merge Conflict

Here's a step-by-step guide to resolving a merge conflict:

1. **Identify the Conflict**: When you try to merge and a conflict occurs, Git will tell you which files have conflicts.3 The conflict markers (<<<<<<<, =======, >>>>>>>) will be inserted into the file.4
   * <<<<<<< HEAD: This marks the beginning of the changes from the current branch.5
   * =======: This separates the changes from the two branches.
   * >>>>>>> branch-name: This marks the end of the changes from the branch you are merging.6
2. **Manually Edit the File**: Open the file with the conflict and manually edit the code.7 You must delete the conflict markers (<<<<<<<, =======, >>>>>>>) and choose which version of the code you want to keep. You can keep both, just one, or a completely new version.
3. **Stage the Changes**: After you have resolved all the conflicts in the file, use git add <filename> to stage the file.8 This tells Git that the conflict has been resolved.
4. **Complete the Merge**: Once all conflicted files are staged, use git commit to finalize the merge.9 Git will automatically create a commit message for you, which you can edit if you like.

### Tips for a Smooth Resolution

* **Communicate with your team**: If you're working with others, it's a good idea to talk to the person who made the conflicting changes to understand their code.
* **Use a graphical merge tool**: Many code editors and IDEs have built-in tools or plugins that make it easier to visualize and resolve conflicts.10
* **Merge frequently**: Merging changes from the main branch into your feature branch often can help you avoid major conflicts.

**Output:**

ARVINTH@LAPTOP-KOK8JOQ1 MINGW64 ~/Downloads/GitDemo (master)

$ git status

git branch

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

On branch master

nothing to commit, working tree clean

GitWork

feature/user-login

\* master

\* bfa366e (**HEAD** -> **master**) WIP: save changes before merge

\* 0a9bf5b Update hello.xml in GitWork

\* 43d1373 Merge branch 'GitWork' Merge conflict

|\

| \* 04b00b3 (**GitWork**) Added hello.xml in GitWork branch

\* | 4710952 Master branch change to hello.xml before merge

\* | 2597eda Added hello.xml in master branch

|/

| \* 23d751d (**feature/user-login**) Added login feature

|/

\* b59d11d (**origin/master**) Initial commit

ARVINTH@LAPTOP-KOK8JOQ1 MINGW64 ~/Downloads/GitDemo (master)

$ git checkout master

git status

Already on 'master'

On branch master

nothing to commit, working tree clean

ARVINTH@LAPTOP-KOK8JOQ1 MINGW64 ~/Downloads/GitDemo (master)

$ git add .

git commit -m "WIP: save changes before merge"

On branch master

nothing to commit, working tree clean

ARVINTH@LAPTOP-KOK8JOQ1 MINGW64 ~/Downloads/GitDemo (master)

$ git checkout GitWork

notepad hello.xml git add hello.xml

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

Switched to branch 'GitWork'

On branch GitWork

Untracked files:

(use "git add <file>..." to include in what will be committed)

error.log

log/

nothing added to commit but untracked files present (use "git add" to track)

ARVINTH@LAPTOP-KOK8JOQ1 MINGW64 ~/Downloads/GitDemo (GitWork)

$ git status

git diff hello.xml

On branch GitWork

Untracked files:

(use "git add <file>..." to include in what will be committed)

error.log

log/

nothing added to commit but untracked files present (use "git add" to track)

ARVINTH@LAPTOP-KOK8JOQ1 MINGW64 ~/Downloads/GitDemo (GitWork)

$ git add hello.xml

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

On branch GitWork

Untracked files:

(use "git add <file>..." to include in what will be committed)

error.log

log/

nothing added to commit but untracked files present (use "git add" to track)

ARVINTH@LAPTOP-KOK8JOQ1 MINGW64 ~/Downloads/GitDemo (GitWork)

$ git checkout master

Switched to branch 'master'

ARVINTH@LAPTOP-KOK8JOQ1 MINGW64 ~/Downloads/GitDemo (master)

$ notepad hello.xml

ARVINTH@LAPTOP-KOK8JOQ1 MINGW64 ~/Downloads/GitDemo (master)

$ git add hello.xml

git commit -m "Add hello.xml in master branch (different content)"

[master 32bd75c] Add hello.xml in master branch (different content)

1 file changed, 1 insertion(+), 1 deletion(-)

ARVINTH@LAPTOP-KOK8JOQ1 MINGW64 ~/Downloads/GitDemo (master)

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

\* 32bd75c (**HEAD** -> **master**) Add hello.xml in master branch (different content)

\* bfa366e WIP: save changes before merge

\* 0a9bf5b Update hello.xml in GitWork

\* 43d1373 Merge branch 'GitWork' Merge conflict

|\

| \* 04b00b3 (**GitWork**) Added hello.xml in GitWork branch

\* | 4710952 Master branch change to hello.xml before merge

\* | 2597eda Added hello.xml in master branch

|/

| \* 23d751d (**feature/user-login**) Added login feature

|/

\* b59d11d (**origin/master**) Initial commit

ARVINTH@LAPTOP-KOK8JOQ1 MINGW64 ~/Downloads/GitDemo (master)

$ git diff master..GitWork -- hello.xml

**diff --git a/hello.xml b/hello.xml**

**index bda6d15..3d4e617 100644**

**--- a/hello.xml**

**+++ b/hello.xml**

@@ -1 +1 @@

-<message>Modifed in Master branch</message>

+<message>Updated in GitWork branch</message>

ARVINTH@LAPTOP-KOK8JOQ1 MINGW64 ~/Downloads/GitDemo (master)

$ git checkout master

git merge GitWork

Already on 'master'

Already up to date.

