

# Assignment 03:

## Feature Branches and Hotfixes

Create a 'hotfix' branch to fix an issue in the main code.  
Merge the 'hotfix' branch into 'main' ensuring that the issue is resolved.

```
MINGW64:/c/Users/user/Documents/Repository
$ git checkout hotfix
Switched to branch 'hotfix'

Vaishnavi J@DESKTOP-73BD0TJ MINGW64 ~/Documents/Repository (hotfix)
$ git ls-files
example.txt
mintty.2024-05-18_20-50-49.png
one.txt

Vaishnavi J@DESKTOP-73BD0TJ MINGW64 ~/Documents/Repository (hotfix)
$ vim one.txt

Vaishnavi J@DESKTOP-73BD0TJ MINGW64 ~/Documents/Repository (hotfix)
$ add .
bash: add: command not found

Vaishnavi J@DESKTOP-73BD0TJ MINGW64 ~/Documents/Repository (hotfix)
$ git add .
warning: in the working copy of 'one.txt', LF will be replaced by CRLF the next time Git touches it

Vaishnavi J@DESKTOP-73BD0TJ MINGW64 ~/Documents/Repository (hotfix)
$ git commit -m "added file one.txt"
[hotfix f8e198c] added file one.txt
1 file changed, 1 insertion(+)

Vaishnavi J@DESKTOP-73BD0TJ MINGW64 ~/Documents/Repository (hotfix)
$ git checkout main
Switched to branch 'main'

Vaishnavi J@DESKTOP-73BD0TJ MINGW64 ~/Documents/Repository (main)
$ git merge hotfix
Updating fd94570..f8e198c
Fast-forward
 one.txt | 1 +
1 file changed, 1 insertion(+)
```

### 1. Create a Hotfix Branch:

Fix the Issue:

Make the necessary changes to fix the issue in your code.

## 2. Commit the Changes:

This stages and commits the changes you made to the 'hotfix' branch.

## 3. Switch back to the Main Branch:

## 4. Merge Hotfix into Main:

This command merges the changes from the 'hotfix' branch into the 'main' branch. If there are no conflicts, Git will automatically perform a fast-forward merge.

## 5. Resolve Conflicts (if any):

If there are conflicts during the merge, you'll need to resolve them manually. Git will prompt you to resolve conflicts and commit the changes.

## 6. Push Changes:

```
git push origin main
```

This command pushes the changes in the 'main' branch (including the hotfix) to the remote repository.