**Assignment 3** : **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**.

**Ans :**

Certainly! Here's how you can create a 'hotfix' branch to fix an issue in the main code and then merge it into the 'main' branch:

1. **Create a new branch named 'hotfix'** using the git branch command:

bash

git branch hotfix

This command creates a new branch named 'hotfix' based on the current branch (which could be 'master' or 'main').

1. **Switch to the 'hotfix' branch** using the git checkout command:

bash

git checkout hotfix

This command switches your working directory to the 'hotfix' branch.

1. **Make the necessary fixes** to resolve the issue in the codebase.
2. **Add the changes to the staging area** using the git add command:

bash

git add .

This command stages all changes in the current directory and its subdirectories for commit.

1. **Commit the changes** to the 'hotfix' branch using the git commit command:

bash

git commit -m "Fixed issue in main code"

This command commits the staged changes to the 'hotfix' branch with an appropriate commit message.

1. **Switch back to the 'main' branch** using the git checkout command:

bash

git checkout main

This command switches your working directory back to the 'main' branch.

1. **Merge the 'hotfix' branch into the 'main' branch** using the git merge command:

bash

git merge hotfix

This command merges the changes from the 'hotfix' branch into the 'main' branch.

1. **Resolve any merge conflicts**, if they occur, by editing the conflicted files, staging the changes, and completing the merge using git commit.

Now, the issue has been fixed in the 'hotfix' branch, and the changes have been merged into the 'main' branch, ensuring that the issue is resolved in the main codebase.