**WEEK 8 HANDSON : GIT**

**TASK 3 :**

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

* Construct a branch, do some changes in the branch, and merge it with master (or trunk)

**TASK AND OUTPUTS** :

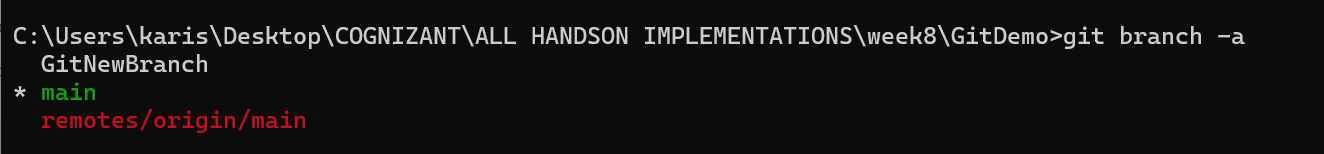
**Branching:**

* A branch is an independent line of development in Git.
* Allows you to work on new features or bug fixes without affecting the main codebase.
* Default branch is usually main or master.

1. Create a new branch **“GitNewBranch”.**

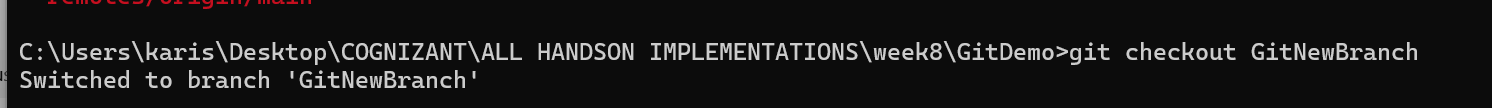


1. List all the local and remote branches available in the current trunk. Observe the “\*” mark which denote the current pointing branch.

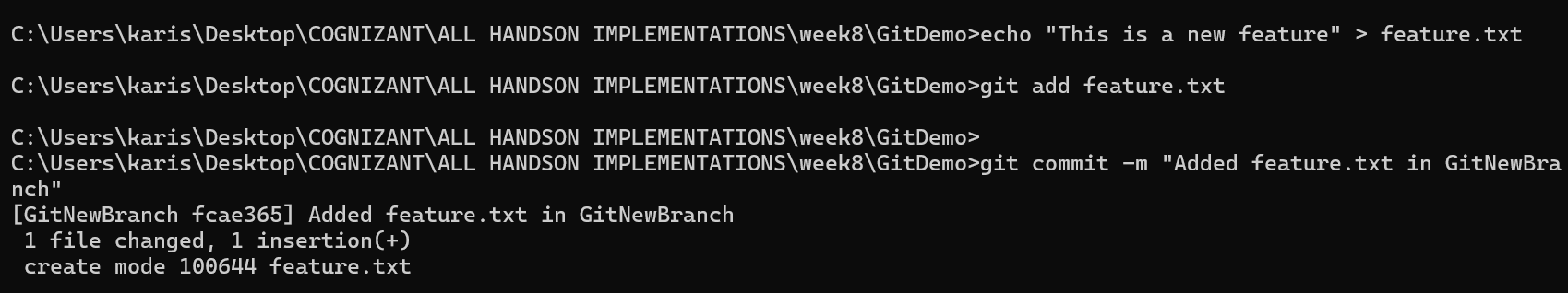


1. Switch to the newly created branch. Add some files to it with some contents.

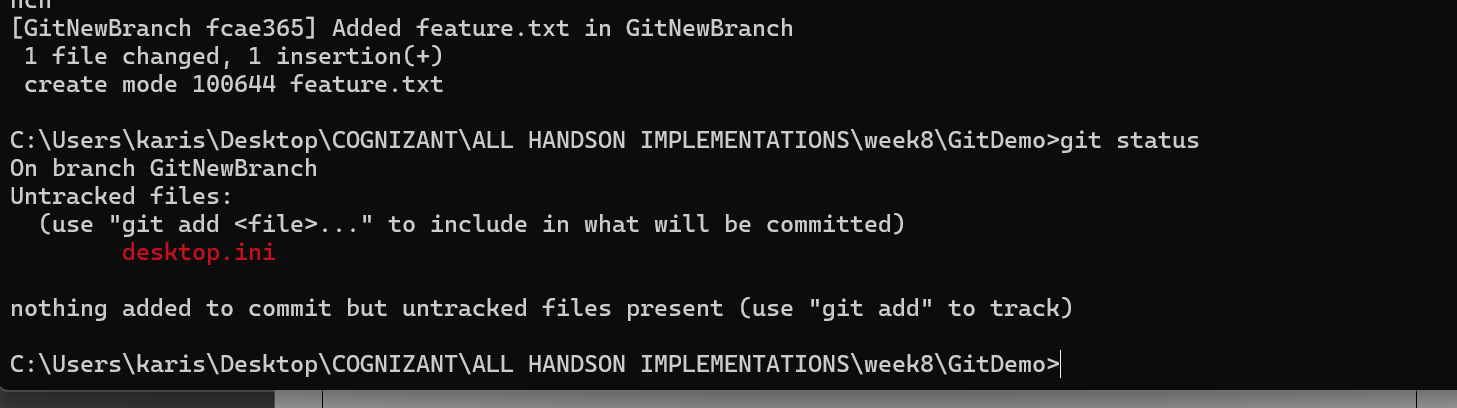
git checkout GitNewBranch

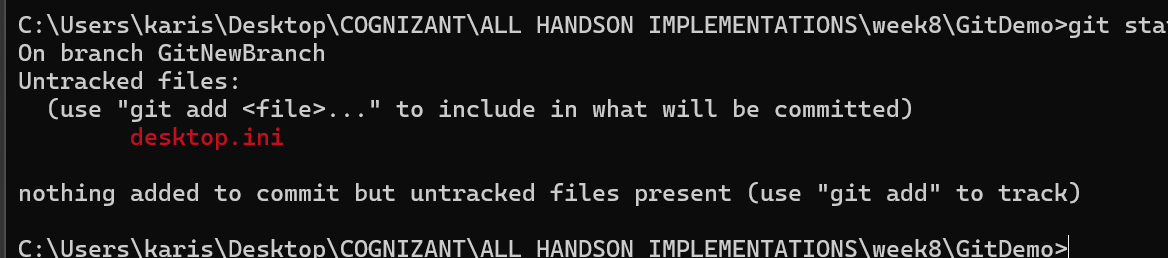


1. Commit the changes to the branch.



1. Check the status with **“git status”** command.





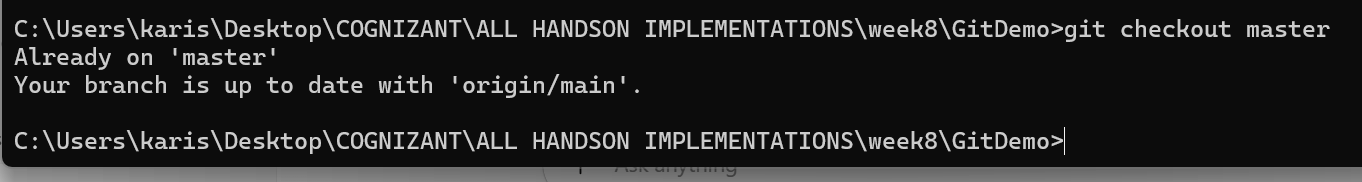
**Merging:**

1. **Merging combines changes from one branch into another.**
2. **Can be:**

* **Fast-forward merge (simple pointer move)**
* **Three-way merge (creates a merge commit when histories diverge)**

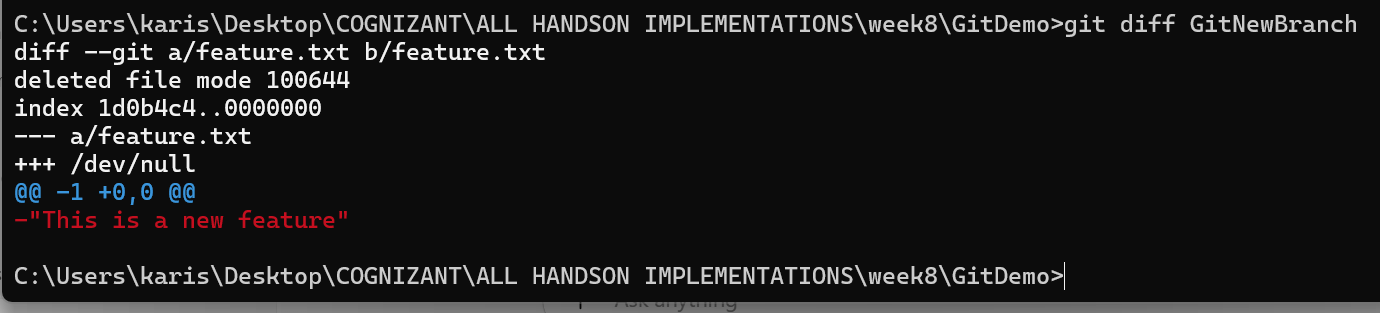
1. **Conflicts occur when changes overlap; these need manual resolution.**
2. Switch to the master

Git checkout master



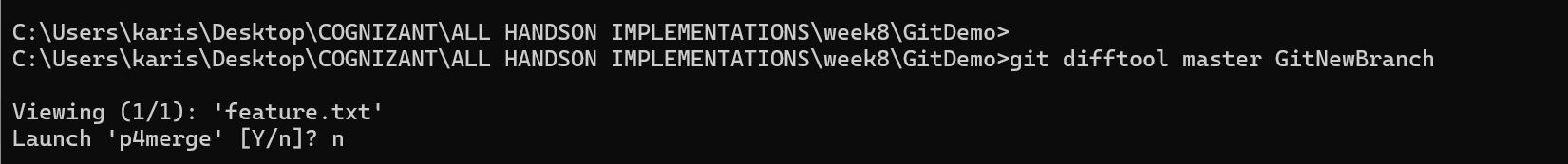
1. List out all the differences between trunk and branch. These provide the differences in command line interface.

git diff GitNewBranch



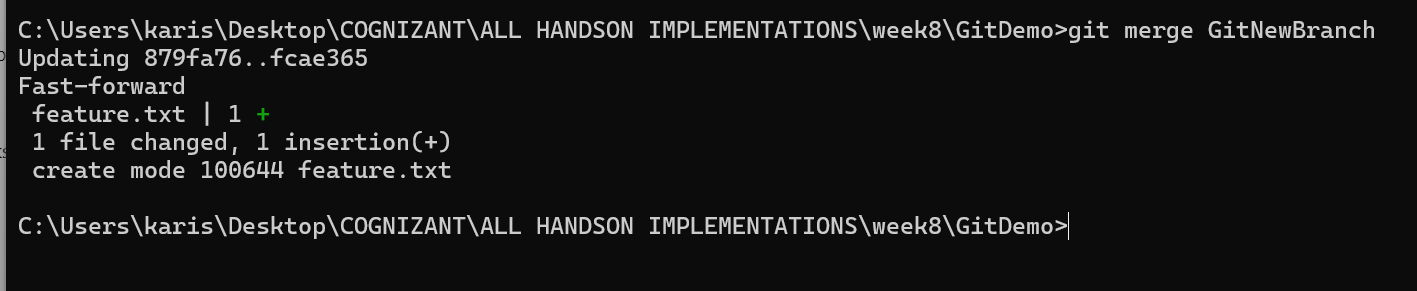
1. List out all the visual differences between master and branch using **P4Merge tool**.

git difftool master GitNewBranch



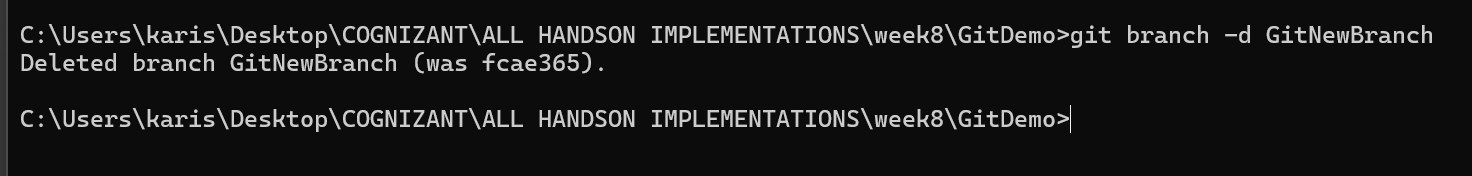
1. Merge the source branch to the trunk

git merge GitNewBranch



1. Observe the logging after merging using **“git log –oneline –graph –decorate”**
2. Delete the branch after merging with the trunk and observe the git status.

**git branch -d GitNewBranch**



Git status

