

# DN 4.0 JAVA FSE SOLUTIONS – WEEK 8

## Skill: GIT

### Hands-On 3

#### Objective

This document outlines the steps taken to create a new Git branch, make changes within it, and then merge those changes back into the master branch.

#### Branching

The following steps were performed to create and work on a new branch.

- **Create a new branch:** I created a new branch named GitNewBranch from the master branch.
  - **Command:**
  - `git branch GitNewBranch`
- **List all branches:** I used the -a flag to list all local and remote branches. The \* showed that I was still on the master branch.
  - **Command:**
  - `git branch -a`

#### Output:

```
$ git branch -a
GitNewBranch
* master
remotes/origin/master
```

- **Switch to branch and add a file:** I switched to the new branch and created a file named newfile.txt to add some content.
  - **Commands:**
  - `git checkout GitNewBranch`
  - `echo "This is a new file for the new branch." > newfile.txt`
  - `git add .`

#### Output:

```
$ git checkout GitNewBranch
echo "This is a new file for the new branch." > newfile.txt
git add .
Switched to branch 'GitNewBranch'
warning: in the working copy of 'newfile.txt', LF will be replaced by CRLF the n
ext time Git touches it
```

- **Commit changes:** I committed the new file to the GitNewBranch with a descriptive message.
  - **Command:** `git commit -m "Add newfile.txt to GitNewBranch"`

### Output:

```
$ git commit -m "Add newfile.txt to GitNewBranch"
[GitNewBranch 41b3847] Add newfile.txt to GitNewBranch
1 file changed, 1 insertion(+)
create mode 100644 newfile.txt
```

- **Check status:** I verified that the working directory was clean after the commit.
  - **Command:**
  - `git status`

### Output:

```
$ git status
On branch GitNewBranch
nothing to commit, working tree clean
```

## Merging

After making changes on the GitNewBranch, I merged them back into the master branch.

- **Switch to master:** I switched back to the master branch, which is the target for the merge.
  - **Command:** `git checkout master`

### Output:

```
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
```

- **List command-line differences:** I used `git diff` to see the changes between the two branches in the terminal.
  - **Command:** `git diff master GitNewBranch`

### Output:

```
$ git diff master GitNewBranch
diff --git a/newfile.txt b/newfile.txt
new file mode 100644
index 0000000..7deb223
--- /dev/null
+++ b/newfile.txt
@@ -0,0 +1 @@
+This is a new file for the new branch.
```

- **List visual differences:** I used the `difftool` command to open P4Merge for a visual comparison of the changes.
  - **Command:** `git difftool master GitNewBranch`

### Output:

```
$ git difftool master GitNewBranch

Viewing (1/1): 'newfile.txt'
Launch 'p4merge' [Y/n]? y
```

- **Merge the branches:** I merged GitNewBranch into the master branch.

- **Command:** git merge GitNewBranch

**Output:**

```
$ git merge GitNewBranch
Updating 580210e..41b3847
Fast-forward
 newfile.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 newfile.txt
```

- **Observe the log:** I used git log to see the commit history, including the merge.

- **Command:** git log --oneline --graph --decorate

**Output:**

```
$ git log --oneline --graph --decorate
* 41b3847 (HEAD -> master, GitNewBranch) Add newfile.txt to GitNewBranch
* 580210e (origin/master) Add .gitignore to ignore log files and folder
* 309b924 First commit
```

- **Delete the branch:** After a successful merge, I deleted the GitNewBranch locally.

- **Command:** git branch -d GitNewBranch

**Output:**

```
$ git branch -d GitNewBranch
Deleted branch GitNewBranch (was 41b3847).
```

```
$ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```