

**Step 1 : Create a New Local Repository.**

1. Create a project directory with the name of "Your\_employee\_id - Your Name" .
2. Use the appropriate git command to initialize your project directory as a git repository.
3. Create 5 .java files and 5 .html files inside this repository and verify the git status.
4. Add all the .java files into the staging area and once again verify the status.
5. Now make a commit with the message "Java Commit".
6. Now add all the .html files into the staging area and make a commit with the message "Html Commit".
7. Display all the version history along with the commit messages and timestamp.

**Step 2: Create a New Branch in the Local Repository.**

1. Create a New Branch with the name "My-Branch" and confirm the branch creation.
2. Create two text files inside this branch and verify the status.
3. Add these 2 files into the staging area and commit with the message "My-Branch-Commit" and verify it.
4. And merge "My-Branch" with master branch and confirm it.

**Step 3: Overcome the merge conflict in the Local Repository.**

1. Create, add and commit a file "File1.txt" into the local repository.
2. Create two New Branches with the name "My-Branch1", "My-Branch2" and confirm the branch creation.
3. Add additional contents in File1 .txt file inside this branch1 and verify the status.
4. Add this file into the staging area and commit with the message "My-Branch1-Commit" and verify it.
5. And merge "My-Branch1" with master branch and confirm it.
6. Similarly checkout to My-Branch2 and add additional contents in File1 .txt file inside this branch2 and verify the status.
7. Add this file into the staging area and commit with the message "My-Branch2-Commit" and verify it.
8. And merge "My-Branch2" with master branch and check for the output.

9. If any conflicts, rectify and add it to the staging area and commit.

**Step 4: Restoring the files from the Git repository and undoing the recent commit.**

1. Remove File1.txt from the present working directory.
2. Check the status of git.
3. Add all the files to the staging area and commit with the message of "File1 Deleted".
4. Now check the list of files in git.
5. Undo the recent commit and then check the list of files in git.
6. Restore the deleted from git repository to the present working directory

**Ans:**

**Step 1**

```
MINGW64/d/12118-Prem
premi.mariyappan@RLP1860 MINGW64 /d
$ cd D:

premi.mariyappan@RLP1860 MINGW64 /d
$ mkdir 12118-Prem

premi.mariyappan@RLP1860 MINGW64 /d
$ cd D:12118-Prem

premi.mariyappan@RLP1860 MINGW64 /d/12118-Prem
$ git init
Initialized empty Git repository in D:/12118-Prem/.git/

premi.mariyappan@RLP1860 MINGW64 /d/12118-Prem (master)
$ cat > File1.java
System.out.println("first Java File");

premi.mariyappan@RLP1860 MINGW64 /d/12118-Prem (master)
$ cat > File2.java
System.out.println("Second Java File");

premi.mariyappan@RLP1860 MINGW64 /d/12118-Prem (master)
$ cat > File3.java
System.out.println("Third Java File");

premi.mariyappan@RLP1860 MINGW64 /d/12118-Prem (master)
$ cat > File4.java
System.out.println("Fourth Java File");

premi.mariyappan@RLP1860 MINGW64 /d/12118-Prem (master)
$ cat > File5.java
System.out.println("Fifth Java File");
```

```
prem.mariyappan@RLP1860 MINGW64 /d/12118-Prem (master)
$ cat > File1.html
1st HTML file

prem.mariyappan@RLP1860 MINGW64 /d/12118-Prem (master)
$ cat > File2.html
2nd HTML file

prem.mariyappan@RLP1860 MINGW64 /d/12118-Prem (master)
$ cat > File3.html
3rd HTML file

prem.mariyappan@RLP1860 MINGW64 /d/12118-Prem (master)
$ cat > File4.html
4th HTML file

prem.mariyappan@RLP1860 MINGW64 /d/12118-Prem (master)
$ cat > File5.html
5th HTML file
git status

prem.mariyappan@RLP1860 MINGW64 /d/12118-Prem (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    File1.html
    File1.java
    File2.html
    File2.java
    File3.html
    File3.java
    File4.html
```

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

prem.mariyappan@RLP1860 MINGW64 /d/12118-Prem (master)
$ git add File1.java File2.java File3.java File4.java File5.java
warning: in the working copy of 'File1.java', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'File2.java', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'File3.java', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'File4.java', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'File5.java', LF will be replaced by CRLF the next time Git touches it

prem.mariyappan@RLP1860 MINGW64 /d/12118-Prem (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   File1.java
    new file:   File2.java
    new file:   File3.java
    new file:   File4.java
    new file:   File5.java

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    File1.html
    File2.html
    File3.html
    File4.html
    File5.html

prem.mariyappan@RLP1860 MINGW64 /d/12118-Prem (master)
$ git commit -m "Java Commit"
[master (root-commit) c490692] Java Commit
5 files changed, 9 insertions(+)
create mode 100644 File1.java
create mode 100644 File2.java
create mode 100644 File3.java
create mode 100644 File4.java
create mode 100644 File5.java
```

```
MINGW64:/d/12118-Prem
premi@mariyappan@RLP1860 MINGW64 /d/12118-Prem (master)
$ git commit -m "Java Commit"
[master (root-commit) c490692] Java Commit
5 files changed, 9 insertions(+)
create mode 100644 File1.java
create mode 100644 File2.java
create mode 100644 File3.java
create mode 100644 File4.java
create mode 100644 File5.java

premi@mariyappan@RLP1860 MINGW64 /d/12118-Prem (master)
$ git add File1.html File2.html File3.html File4.html File5.html
warning: in the working copy of 'File1.html', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'File2.html', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'File3.html', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'File4.html', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'File5.html', LF will be replaced by CRLF the next time Git touches it

premi@mariyappan@RLP1860 MINGW64 /d/12118-Prem (master)
$ git commit -m "HTML commit"
[master 9154ae8] HTML commit
5 files changed, 7 insertions(+)
create mode 100644 File1.html
create mode 100644 File2.html
create mode 100644 File3.html
create mode 100644 File4.html
create mode 100644 File5.html

premi@mariyappan@RLP1860 MINGW64 /d/12118-Prem (master)
$ git log
commit 9154ae88978e6787e78b1e3bbce453c83154d5d7 (HEAD -> master)
Author: premi <premi4314@gmail.com>
Date: Sat Dec 16 10:29:26 2023 +0530

    HTML commit

commit c490692145306cfe19df8842bf5cb29360c19fa
Author: premi <premi4314@gmail.com>
Date: Sat Dec 16 10:27:34 2023 +0530

    Java Commit
```

## Step 2

```
MINGW64:/d/Step2
premi@mariyappan@RLP1860 MINGW64 /d/Step2 (master)
$ git branch -a
* My-Branch
* master

premi@mariyappan@RLP1860 MINGW64 /d/Step2 (master)
$ git ls
git: 'ls' is not a git command. See 'git --help'.

The most similar command is
  ls

premi@mariyappan@RLP1860 MINGW64 /d/Step2 (master)
$ ls
File.txt

premi@mariyappan@RLP1860 MINGW64 /d/Step2 (master)
$ switch My-Branch
bash: switch: command not found

premi@mariyappan@RLP1860 MINGW64 /d/Step2 (master)
$ git switch My-Branch
Switched to branch 'My-Branch'

premi@mariyappan@RLP1860 MINGW64 /d/Step2 (My-Branch)
$ ls
File.txt

premi@mariyappan@RLP1860 MINGW64 /d/Step2 (My-Branch)
$ cat > File1.txt
premi

premi@mariyappan@RLP1860 MINGW64 /d/Step2 (My-Branch)
$ git add File1.txt
warning: in the working copy of 'File1.txt', LF will be replaced by CRLF the next time Git touches it

premi@mariyappan@RLP1860 MINGW64 /d/Step2 (My-Branch)
$ cat > File2.txt
premi2

premi@mariyappan@RLP1860 MINGW64 /d/Step2 (My-Branch)
$ git add File2.txt
warning: in the working copy of 'File2.txt', LF will be replaced by CRLF the next time Git touches it

premi@mariyappan@RLP1860 MINGW64 /d/Step2 (My-Branch)
$ git Commit -m "My Branch Committed"
git: 'Commit' is not a git command. See 'git --help'.

The most similar command is
  commit
```

```
MINGW64/d/Step2
commit
premm@mariyappan@RLP1860 MINGW64 /d/Step2 (My-Branch)
$ git commit -m "My Branch committed"
[My-Branch 4e7e447] My Branch committed
2 files changed, 2 insertions(+)
create mode 100644 File1.txt
create mode 100644 File2.txt

premm@mariyappan@RLP1860 MINGW64 /d/Step2 (My-Branch)
$ git status
On branch My-Branch
nothing to commit, working tree clean

premm@mariyappan@RLP1860 MINGW64 /d/Step2 (My-Branch)
$ switch master
bash: switch: command not found

premm@mariyappan@RLP1860 MINGW64 /d/Step2 (My-Branch)
$ git switch master
Switched to branch 'master'

premm@mariyappan@RLP1860 MINGW64 /d/Step2 (master)
$ git merge My-Branch
Updating fb08942..4e7e447
Fast-forward
 File1.txt | 1 +
 File2.txt | 1 +
 file.txt  | 3 +--
3 files changed, 4 insertions(+), 1 deletion(-)
create mode 100644 File1.txt
create mode 100644 File2.txt

premm@mariyappan@RLP1860 MINGW64 /d/Step2 (master)
$ git log
commit 4e7e44752a366ff5299ccb83b8eb1325098bae4 (HEAD -> master, My-Branch)
Author: premm <premm4314@gmail.com>
Date: Sat Dec 16 11:15:52 2023 +0530

    My Branch committed

commit 04149b79a64f339a7cdd636ec34cc7888062e63a
Author: premm <premm4314@gmail.com>
Date: Sat Dec 16 11:11:11 2023 +0530

    Branch 1 committed

commit fb0894209be96f1402bb21f7464cc06212f2d934
Author: premm <premm4314@gmail.com>
Date: Sat Dec 16 11:08:53 2023 +0530

    committed
```

## Step 3:

```
MINGW64/d/News
premm@mariyappan@RLP1860 MINGW64 /d/News (master)
$ ls
File1.txt

premm@mariyappan@RLP1860 MINGW64 /d/News (master)
$ cat > File2.txt
grp

premm@mariyappan@RLP1860 MINGW64 /d/News (master)
$ cat > File3.txt
Bala
C

premm@mariyappan@RLP1860 MINGW64 /d/News (master)
$ git add File2.txt File3.txt
warning: in the working copy of 'File2.txt', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'File3.txt', LF will be replaced by CRLF the next time Git touches it

premm@mariyappan@RLP1860 MINGW64 /d/News (master)
$ rm File1.txt

premm@mariyappan@RLP1860 MINGW64 /d/News (master)
$ git commit -m "File1.txt deletd"
[master 4520fbff] File1.txt deletd
2 files changed, 4 insertions(+), 2 deletions(-)

premm@mariyappan@RLP1860 MINGW64 /d/News (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    deleted:    File
    deleted:    File1.txt

no changes added to commit (use "git add" and/or "git commit -a")

premm@mariyappan@RLP1860 MINGW64 /d/News (master)
$ git log
commit 4520fbff1502be22e686d0ddf19adde683f01b5 (HEAD -> master)
Author: premm <premm4314@gmail.com>
Date: Sat Dec 16 11:55:11 2023 +0530

    File1.txt deletd
```

```
MINGW64/d/News
Author: prem <premm4314@gmail.com>
Date: Sat Dec 16 11:55:11 2023 +0530

File1.txt deletd

commit f1245cd4158e4db7aa1dbffc384dbd34d7de940a
Author: prem <premm4314@gmail.com>
Date: Sat Dec 16 11:49:08 2023 +0530

Deleted File1.txt

commit 686af3945778bb25bc2d2aab0b107e75b9304062
Author: prem <premm4314@gmail.com>
Date: Sat Dec 16 11:44:00 2023 +0530

File1.txt Deletd

commit 6d5314700ce92c3d7dcc41bcbfbfc79cc07cbed8
Author: prem <premm4314@gmail.com>
Date: Sat Dec 16 11:40:33 2023 +0530

committed successfully

premm.mariyappan@RLP1860 MINGW64 /d/News (master)
$ git reset --hard HEAD~1
HEAD is now at f1245cd Deleted File1.txt

premm.mariyappan@RLP1860 MINGW64 /d/News (master)
$ ls
File File1.txt File2.txt File3.txt

premm.mariyappan@RLP1860 MINGW64 /d/News (master)
$ git status
On branch master
nothing to commit, working tree clean

premm.mariyappan@RLP1860 MINGW64 /d/News (master)
$ git restore File1.txt

premm.mariyappan@RLP1860 MINGW64 /d/News (master)
$ git status
On branch master
nothing to commit, working tree clean

premm.mariyappan@RLP1860 MINGW64 /d/News (master)
$ git ls-files
File
File1.txt
File2.txt
File3.txt

premm.mariyappan@RLP1860 MINGW64 /d/News (master)
$ |
```

## Step 4:

```
MINGW64/d/Step4

premm.mariyappan@RLP1860 MINGW64 /d/News (master)
$ ls
File File1.txt File2.txt File3.txt

premm.mariyappan@RLP1860 MINGW64 /d/News (master)
$ cd D:

premm.mariyappan@RLP1860 MINGW64 /d
$ mkdir Step4

premm.mariyappan@RLP1860 MINGW64 /d
$ cd D:Step4

premm.mariyappan@RLP1860 MINGW64 /d/Step4
$ git init
Initialized empty Git repository in D:/Step4/.git/

premm.mariyappan@RLP1860 MINGW64 /d/Step4 (master)
$ cat > File1.txt
Bala
premm

premm.mariyappan@RLP1860 MINGW64 /d/Step4 (master)
$ git add File1.txt
warning: in the working copy of 'File1.txt', LF will be replaced by CRLF the next time Git touches it

premm.mariyappan@RLP1860 MINGW64 /d/Step4 (master)
$ git commit -m"committed successfully"
[master (root-commit) 78f5898] committed successfully
1 file changed, 1 insertion(+)
create mode 100644 File1.txt

premm.mariyappan@RLP1860 MINGW64 /d/Step4 (master)
$ la
bash: la: command not found

premm.mariyappan@RLP1860 MINGW64 /d/Step4 (master)
$ la
bash: la: command not found

premm.mariyappan@RLP1860 MINGW64 /d/Step4 (master)
$ ls
File1.txt

premm.mariyappan@RLP1860 MINGW64 /d/Step4 (master)
$ git branch My-Branch1

premm.mariyappan@RLP1860 MINGW64 /d/Step4 (master)
$ git branch My-Branch2

premm.mariyappan@RLP1860 MINGW64 /d/Step4 (master)
$ git checkout My-Branch1
```

MINGW64/d/Step4

```
premi.mariyappan@RPLP1860 MINGW64 /d/Step4 (master)
$ git checkout My-Branch1
Switched to branch 'My-Branch1'

premi.mariyappan@RPLP1860 MINGW64 /d/Step4 (My-Branch1)
$ vi File1.txt

premi.mariyappan@RPLP1860 MINGW64 /d/Step4 (My-Branch1)
$ git add File1.txt

premi.mariyappan@RPLP1860 MINGW64 /d/Step4 (My-Branch1)
$ git commit -m"Branch1 committed"
[My-Branch1 eba28e8] Branch1 committed
1 file changed, 1 insertion(+), 1 deletion(-)

premi.mariyappan@RPLP1860 MINGW64 /d/Step4 (My-Branch1)
$ git switch master
Switched to branch 'master'

premi.mariyappan@RPLP1860 MINGW64 /d/Step4 (master)
$ git merge My-Branch1
Updating 78fs896..eba28e8
Fast-forward
 File1.txt | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)

premi.mariyappan@RPLP1860 MINGW64 /d/Step4 (master)
$ git switch My-Branch2
Switched to branch 'My-Branch2'

premi.mariyappan@RPLP1860 MINGW64 /d/Step4 (My-Branch2)
$ vi File1.txt\
> AC

premi.mariyappan@RPLP1860 MINGW64 /d/Step4 (My-Branch2)
$ ls
File1.txt

premi.mariyappan@RPLP1860 MINGW64 /d/Step4 (My-Branch2)
$ vi File1.txt

premi.mariyappan@RPLP1860 MINGW64 /d/Step4 (My-Branch2)
$ git add.
git: 'add.' is not a git command. See 'git --help'.

The most similar command is
add

premi.mariyappan@RPLP1860 MINGW64 /d/Step4 (My-Branch2)
$ git add .

premi.mariyappan@RPLP1860 MINGW64 /d/Step4 (My-Branch2)
$ git commit -m"Branch2 committed"
```

