PRN: **121A3008**

Experiment No: 3

AIM: Branching and Merging with Git.

THEORY: GitHub is a global company that provides hosting for software development version control using Git. It is a Subsidiary of Microsoft, which acquired the company in 2018 for $7.5 billion. It offers all of the distributed version control and source code management (SCM) functionality of Git as well as adding its own features. It provides access control and several collaboration features such as bug tracking, feature requests, task management, and wikis for every project.

Commands:

1) Git add command

This command is used to add one or more files to staging (Index) area. $ git add filename

To add more than one file
$ git add*

2) Git commit command

This command changes the head. It records or snapshots the file permanently in the version history with a message.

```
$ git commit -m " Commit Message"
```

3) Git status command

The status command is used to display the state of the working directory and the staging area. It allows you to see which changes have been staged, which haven't, and which files aren?t being tracked by Git. It does not show you any information about the committed project history. For this, you need to use the git log. It also lists the files that you've changed and those you still need to add or commit.

```
$ git status
```

4) Git log Command

This command is used to check the commit history.

```
$ git log
```

5) Git concatenate command

PRN: **121A3008**

This command is simply use to display one or multiple files, $ cat filename

This command is used to add contents to file

$ cat >>filename

This command is used to append lines to file

$ cat > filename

6) git branch command

In Git, a branch is a new/separate version of the main repository. $ git branch branchname

7) git checkout command

The git checkout command is used to **s**witch between branches in a repository. It can also be used to restore files from a specific commit or branch. For example, to switch to a branch named dev, you can use:

```
$ git checkout dev
```

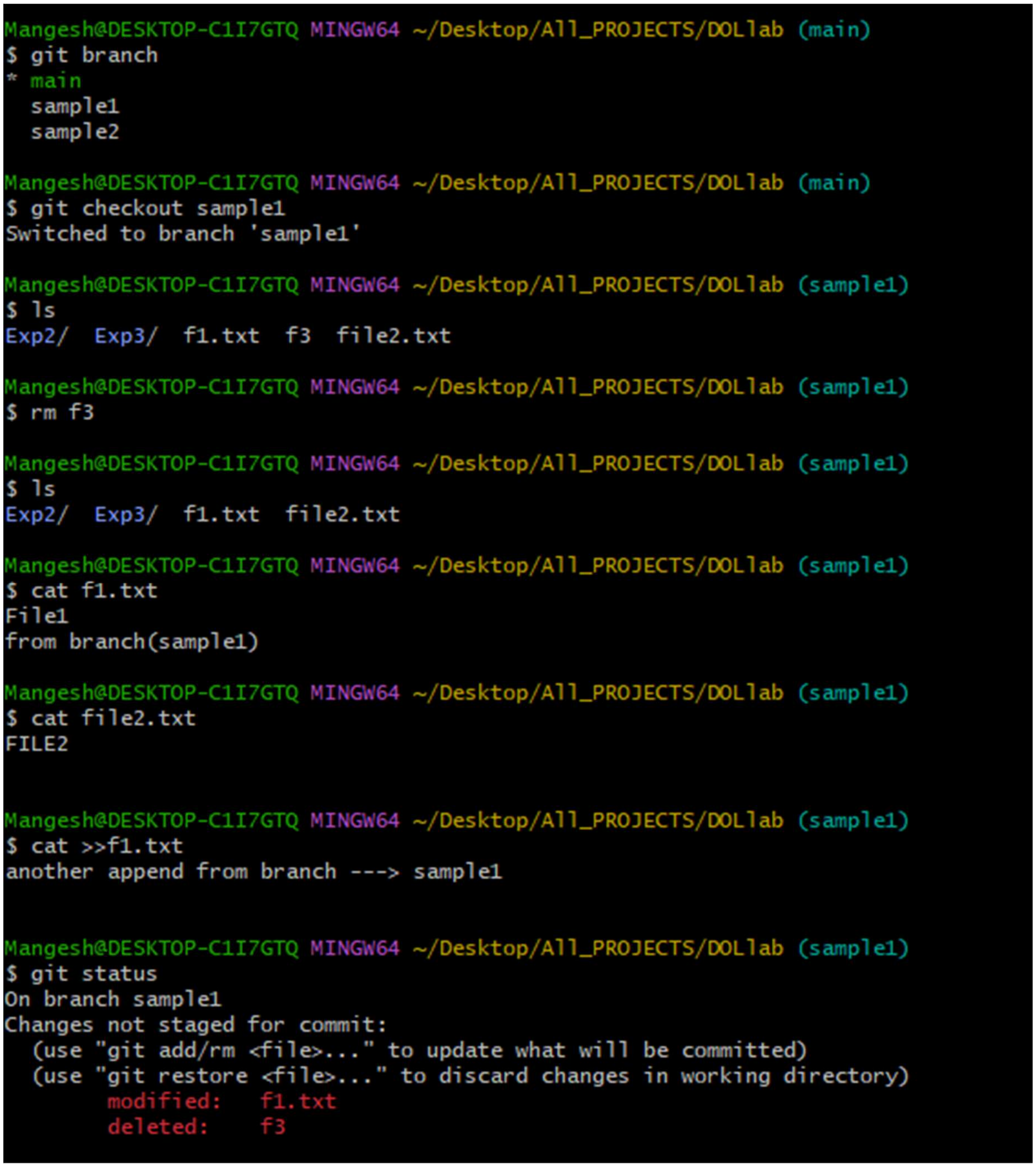
8) git merge command

The git merge command is used to combine two or more branches into one. It takes the changes from one branch and applies them to another branch. For example, if you want to merge branch1 into branch2, you can do:

```
$ git checkout branch2
$ git merge branch1
```

PRN: **121A3008**

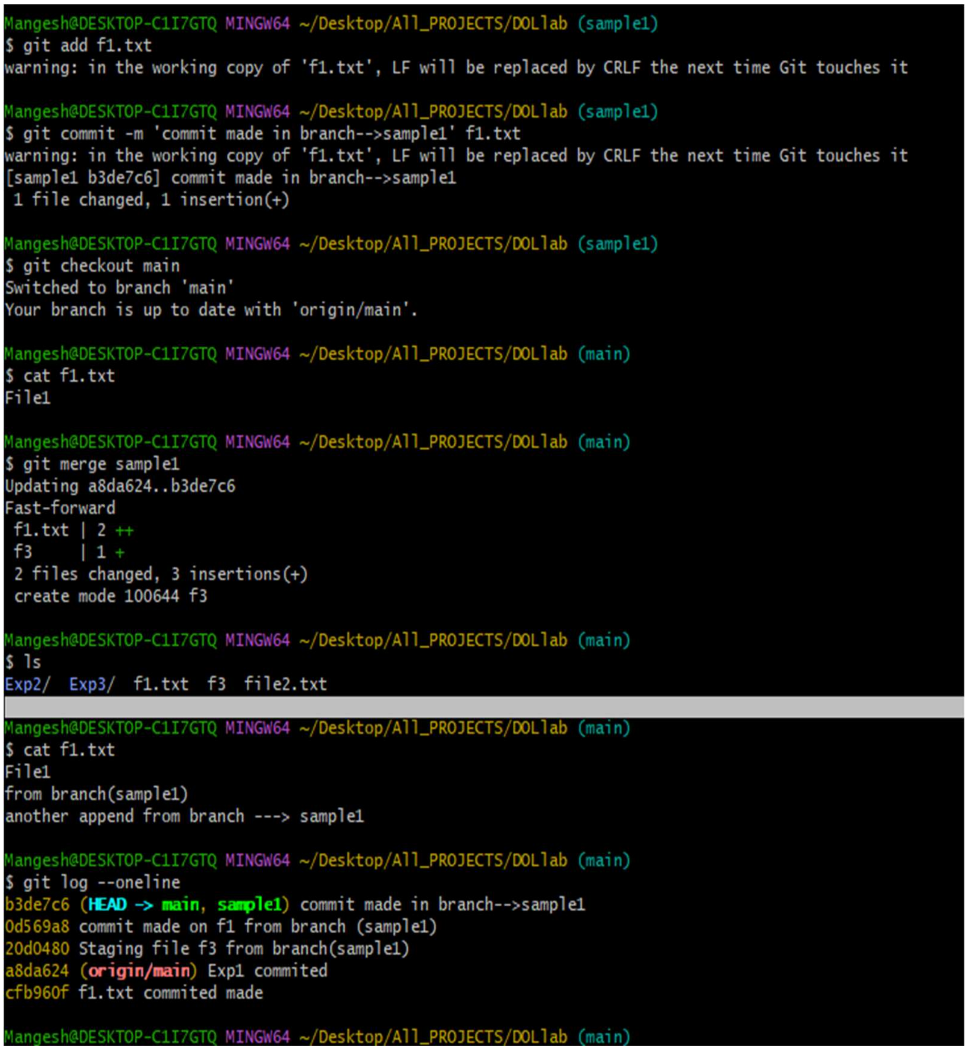
Setting new branch and adding contents to file:



```
Mangesh@DESKTOP-C1I7GTQ MINGW64 ~/Desktop/All_PROJECTS/DOLlab (main)
$ git branch
* main
  sample1
  sample2

Mangesh@DESKTOP-C1I7GTQ MINGW64 ~/Desktop/All_PROJECTS/DOLlab (main)
$ git checkout sample1
Switched to branch 'sample1'

Mangesh@DESKTOP-C1I7GTQ MINGW64 ~/Desktop/All_PROJECTS/DOLlab (sample1)
$ ls
Exp2/  Exp3/  f1.txt  f3  file2.txt

Mangesh@DESKTOP-C1I7GTQ MINGW64 ~/Desktop/All_PROJECTS/DOLlab (sample1)
$ rm f3

Mangesh@DESKTOP-C1I7GTQ MINGW64 ~/Desktop/All_PROJECTS/DOLlab (sample1)
$ ls
Exp2/  Exp3/  f1.txt  file2.txt

Mangesh@DESKTOP-C1I7GTQ MINGW64 ~/Desktop/All_PROJECTS/DOLlab (sample1)
$ cat f1.txt
File1
from branch(sample1)

Mangesh@DESKTOP-C1I7GTQ MINGW64 ~/Desktop/All_PROJECTS/DOLlab (sample1)
$ cat file2.txt
FILE2


Mangesh@DESKTOP-C1I7GTQ MINGW64 ~/Desktop/All_PROJECTS/DOLlab (sample1)
$ cat >>f1.txt
another append from branch ---> sample1


Mangesh@DESKTOP-C1I7GTQ MINGW64 ~/Desktop/All_PROJECTS/DOLlab (sample1)
$ git status
On branch sample1
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)
        modified:   f1.txt
        deleted:    f3
```

PRN: **121A3008**

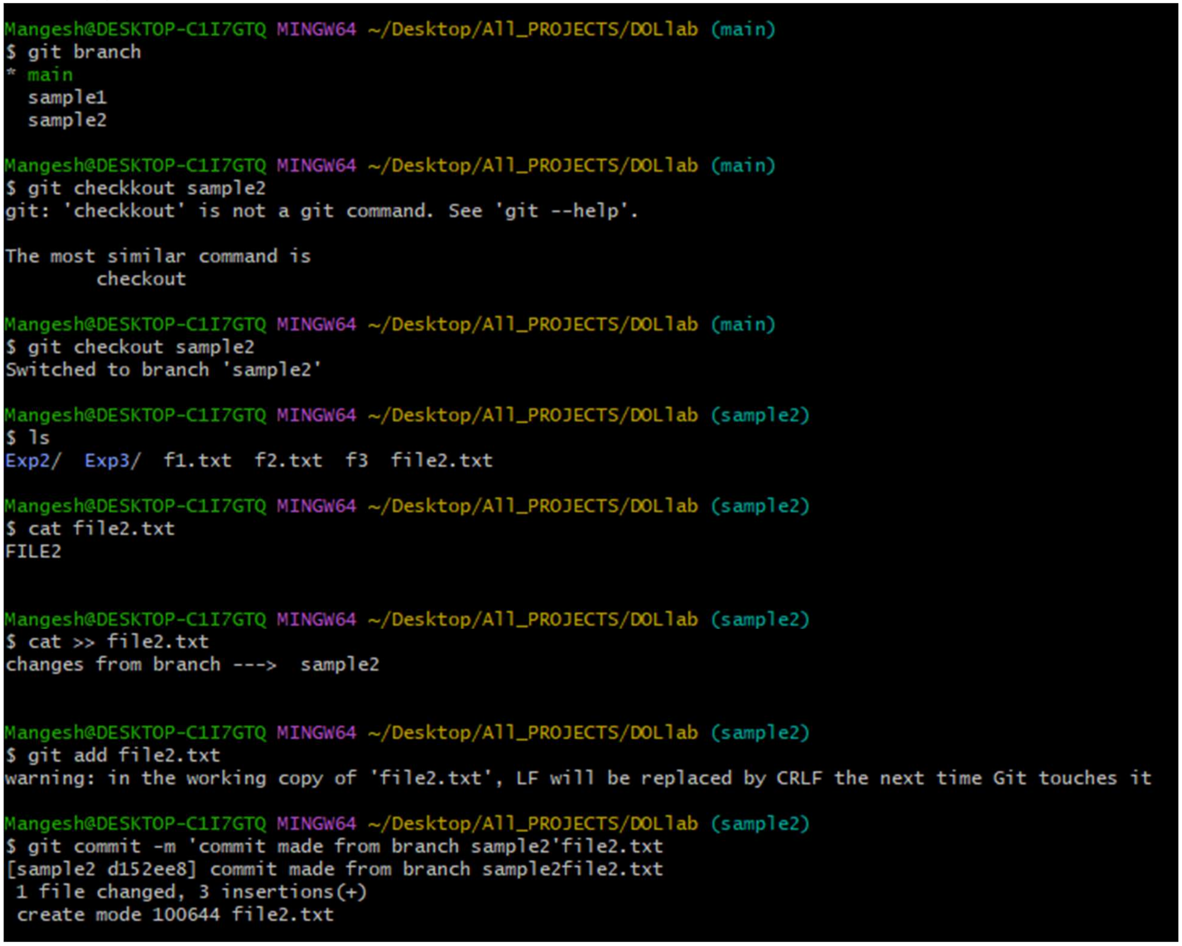
Merging new branch to see changes in main branch:



```
Mangesh@DESKTOP-C1I7GTQ MINGW64 ~/Desktop/All_PROJECTS/DOLlab (sample1)
$ git add f1.txt
warning: in the working copy of 'f1.txt', LF will be replaced by CRLF the next time Git touches it

Mangesh@DESKTOP-C1I7GTQ MINGW64 ~/Desktop/All_PROJECTS/DOLlab (sample1)
$ git commit -m 'commit made in branch-->sample1' f1.txt
warning: in the working copy of 'f1.txt', LF will be replaced by CRLF the next time Git touches it
[sample1 b3de7c6] commit made in branch-->sample1
 1 file changed, 1 insertion(+)

Mangesh@DESKTOP-C1I7GTQ MINGW64 ~/Desktop/All_PROJECTS/DOLlab (sample1)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

Mangesh@DESKTOP-C1I7GTQ MINGW64 ~/Desktop/All_PROJECTS/DOLlab (main)
$ cat f1.txt
File1

Mangesh@DESKTOP-C1I7GTQ MINGW64 ~/Desktop/All_PROJECTS/DOLlab (main)
$ git merge sample1
Updating a8da624..b3de7c6
Fast-forward
 f1.txt | 2 ++
 f3     | 1 +
 2 files changed, 3 insertions(+)
 create mode 100644 f3

Mangesh@DESKTOP-C1I7GTQ MINGW64 ~/Desktop/All_PROJECTS/DOLlab (main)
$ ls
Exp2/  Exp3/  f1.txt  f3  file2.txt

Mangesh@DESKTOP-C1I7GTQ MINGW64 ~/Desktop/All_PROJECTS/DOLlab (main)
$ cat f1.txt
File1
from branch(sample1)
another append from branch ---> sample1

Mangesh@DESKTOP-C1I7GTQ MINGW64 ~/Desktop/All_PROJECTS/DOLlab (main)
$ git log --oneline
b3de7c6 (HEAD -> main, sample1) commit made in branch-->sample1
0d569a8 commit made on f1 from branch (sample1)
20d0480 Staging file f3 from branch(sample1)
a8da624 (origin/main) Exp1 commited
cfb960f f1.txt commited made

Mangesh@DESKTOP-C1I7GTQ MINGW64 ~/Desktop/All_PROJECTS/DOLlab (main)
```

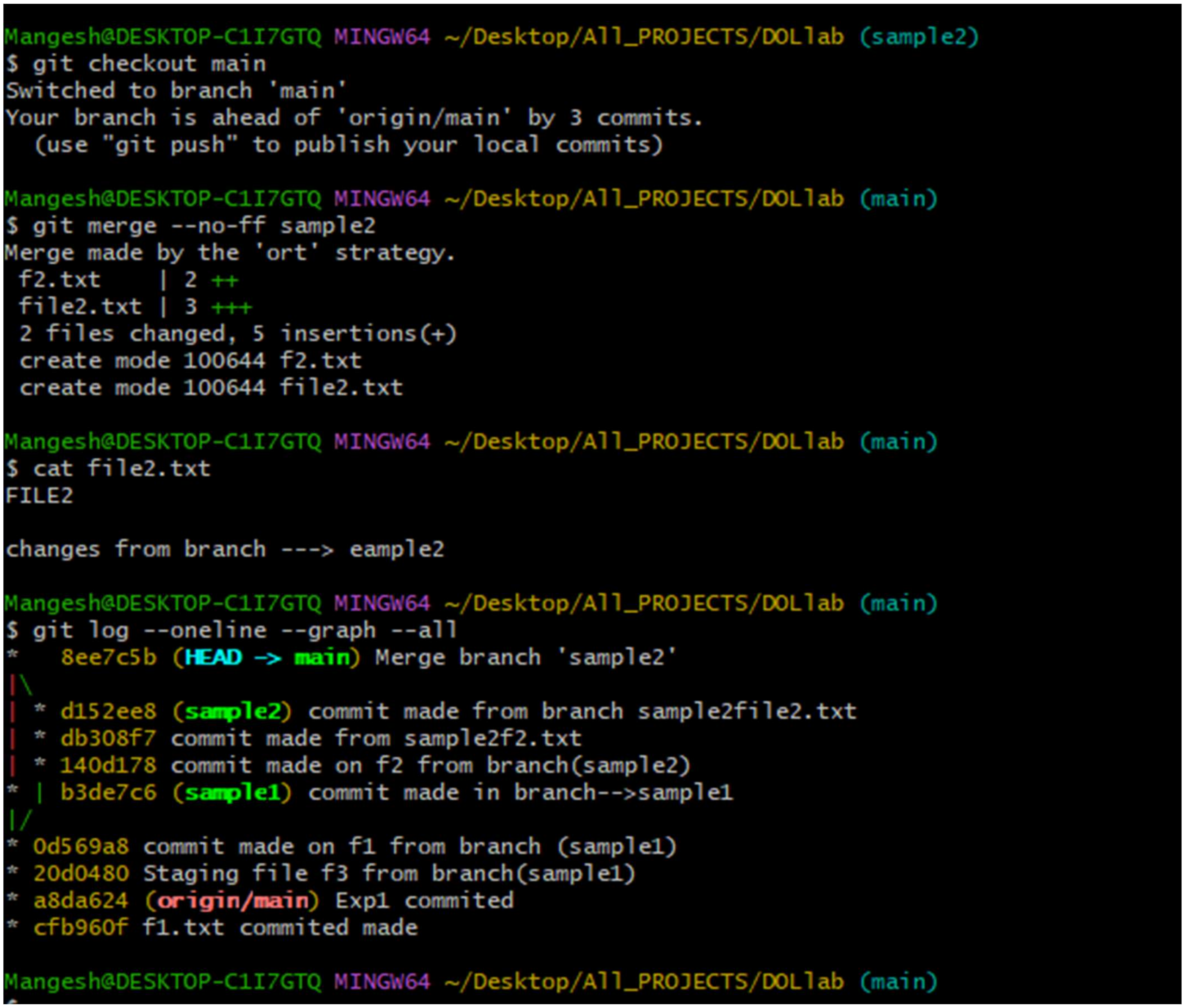
PRN: **121A3008**

Creating new branch and repeat same procedure:



```
Mangesh@DESKTOP-C1I7GTQ MINGW64 ~/Desktop/All_PROJECTS/DOLlab (main)
$ git branch
* main
  sample1
  sample2

Mangesh@DESKTOP-C1I7GTQ MINGW64 ~/Desktop/All_PROJECTS/DOLlab (main)
$ git checkkout sample2
git: 'checkkout' is not a git command. See 'git --help'.

The most similar command is
        checkout

Mangesh@DESKTOP-C1I7GTQ MINGW64 ~/Desktop/All_PROJECTS/DOLlab (main)
$ git checkout sample2
Switched to branch 'sample2'

Mangesh@DESKTOP-C1I7GTQ MINGW64 ~/Desktop/All_PROJECTS/DOLlab (sample2)
$ ls
Exp2/  Exp3/  f1.txt  f2.txt  f3  file2.txt

Mangesh@DESKTOP-C1I7GTQ MINGW64 ~/Desktop/All_PROJECTS/DOLlab (sample2)
$ cat file2.txt
FILE2


Mangesh@DESKTOP-C1I7GTQ MINGW64 ~/Desktop/All_PROJECTS/DOLlab (sample2)
$ cat >> file2.txt
changes from branch --->  sample2


Mangesh@DESKTOP-C1I7GTQ MINGW64 ~/Desktop/All_PROJECTS/DOLlab (sample2)
$ git add file2.txt
warning: in the working copy of 'file2.txt', LF will be replaced by CRLF the next time Git touches it

Mangesh@DESKTOP-C1I7GTQ MINGW64 ~/Desktop/All_PROJECTS/DOLlab (sample2)
$ git commit -m 'commit made from branch sample2'file2.txt
[sample2 d152ee8] commit made from branch sample2file2.txt
 1 file changed, 3 insertions(+)
 create mode 100644 file2.txt
```

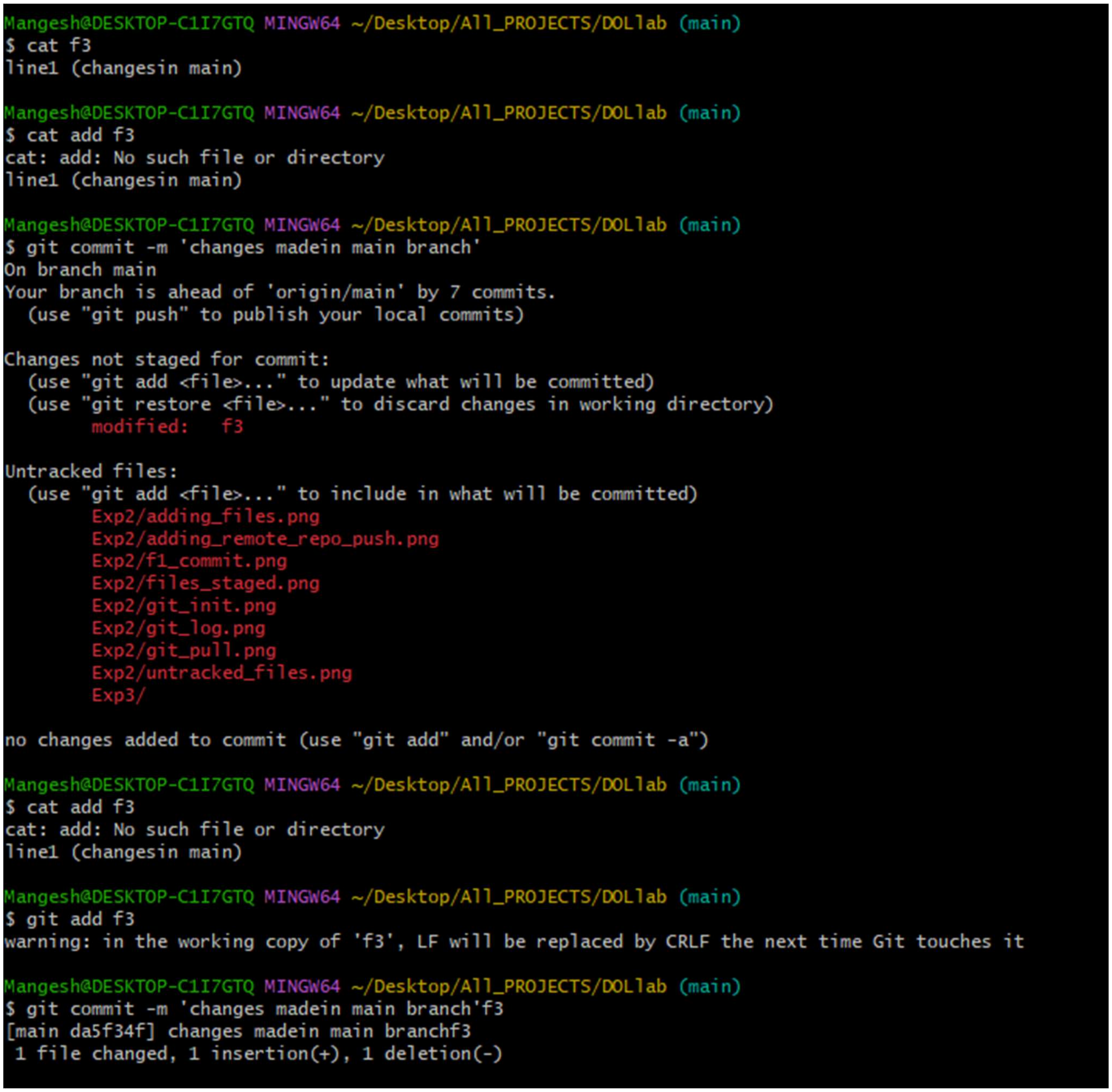
PRN: **121A3008**



```
Mangesh@DESKTOP-C1I7GTQ MINGW64 ~/Desktop/All_PROJECTS/DOLlab (sample2)
$ git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 3 commits.
  (use "git push" to publish your local commits)

Mangesh@DESKTOP-C1I7GTQ MINGW64 ~/Desktop/All_PROJECTS/DOLlab (main)
$ git merge --no-ff sample2
Merge made by the 'ort' strategy.
 f2.txt    | 2 ++
 file2.txt | 3 +++
 2 files changed, 5 insertions(+)
 create mode 100644 f2.txt
 create mode 100644 file2.txt

Mangesh@DESKTOP-C1I7GTQ MINGW64 ~/Desktop/All_PROJECTS/DOLlab (main)
$ cat file2.txt
FILE2

changes from branch ---> eample2

Mangesh@DESKTOP-C1I7GTQ MINGW64 ~/Desktop/All_PROJECTS/DOLlab (main)
$ git log --oneline --graph --all
*   8ee7c5b (HEAD -> main) Merge branch 'sample2'
|\
| * d152ee8 (sample2) commit made from branch sample2file2.txt
| * db308f7 commit made from sample2f2.txt
| * 140d178 commit made on f2 from branch(sample2)
* | b3de7c6 (sample1) commit made in branch-->sample1
|/
* 0d569a8 commit made on f1 from branch (sample1)
* 20d0480 Staging file f3 from branch(sample1)
* a8da624 (origin/main) Exp1 commited
* cfb960f f1.txt commited made

Mangesh@DESKTOP-C1I7GTQ MINGW64 ~/Desktop/All_PROJECTS/DOLlab (main)
```

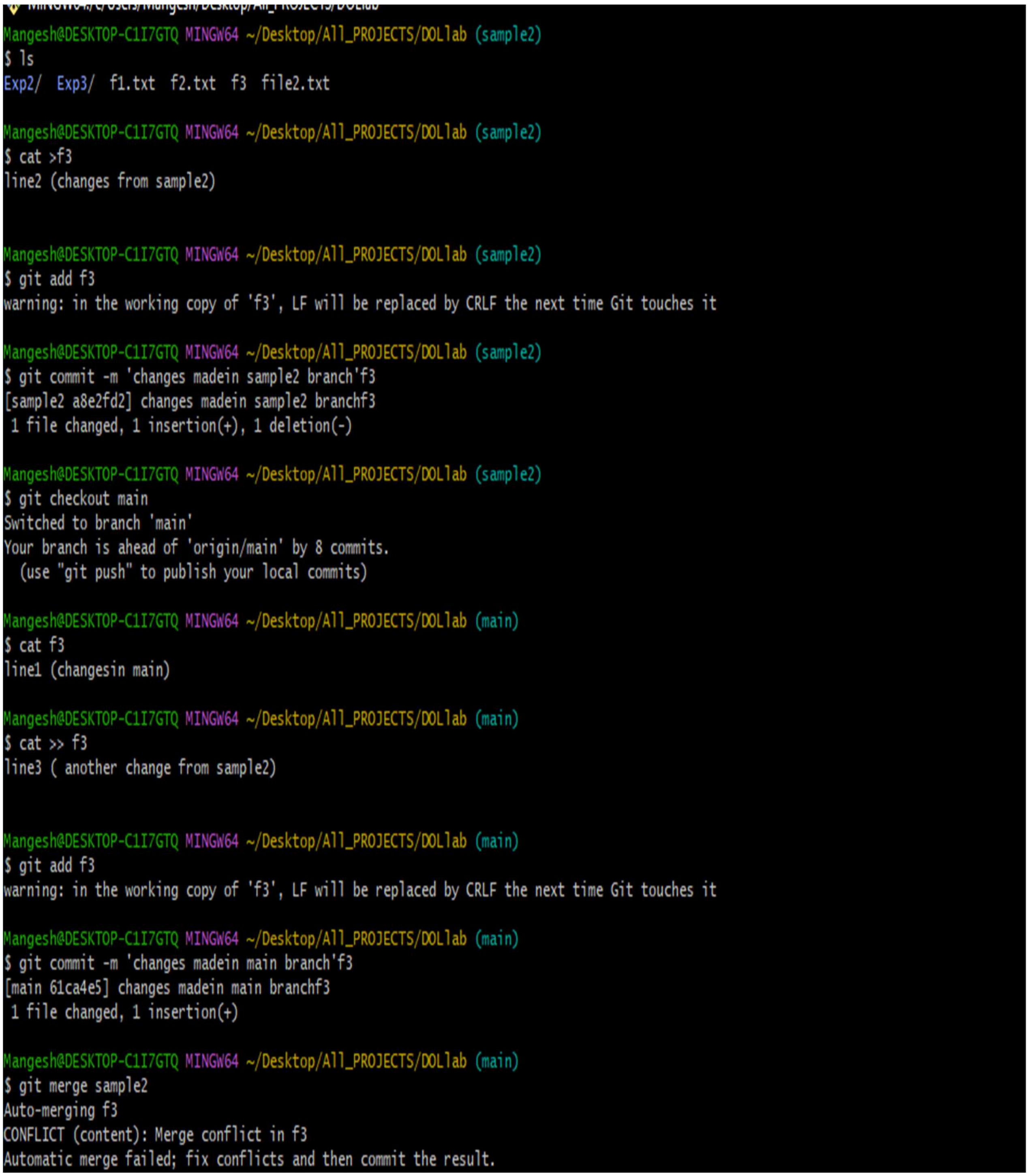
PRN: **121A3008**

Creating new branch and identify merge conflicts:



```
Mangesh@DESKTOP-C1I7GTQ MINGW64 ~/Desktop/All_PROJECTS/DOLlab (main)
$ cat f3
line1 (changesin main)

Mangesh@DESKTOP-C1I7GTQ MINGW64 ~/Desktop/All_PROJECTS/DOLlab (main)
$ cat add f3
cat: add: No such file or directory
line1 (changesin main)

Mangesh@DESKTOP-C1I7GTQ MINGW64 ~/Desktop/All_PROJECTS/DOLlab (main)
$ git commit -m 'changes madein main branch'
On branch main
Your branch is ahead of 'origin/main' by 7 commits.
  (use "git push" to publish your local commits)

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   f3

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Exp2/adding_files.png
        Exp2/adding_remote_repo_push.png
        Exp2/f1_commit.png
        Exp2/files_staged.png
        Exp2/git_init.png
        Exp2/git_log.png
        Exp2/git_pull.png
        Exp2/untracked_files.png
        Exp3/

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

Mangesh@DESKTOP-C1I7GTQ MINGW64 ~/Desktop/All_PROJECTS/DOLlab (main)
$ cat add f3
cat: add: No such file or directory
line1 (changesin main)

Mangesh@DESKTOP-C1I7GTQ MINGW64 ~/Desktop/All_PROJECTS/DOLlab (main)
$ git add f3
warning: in the working copy of 'f3', LF will be replaced by CRLF the next time Git touches it

Mangesh@DESKTOP-C1I7GTQ MINGW64 ~/Desktop/All_PROJECTS/DOLlab (main)
$ git commit -m 'changes madein main branch'f3
[main da5f34f] changes madein main branchf3
 1 file changed, 1 insertion(+), 1 deletion(-)
```

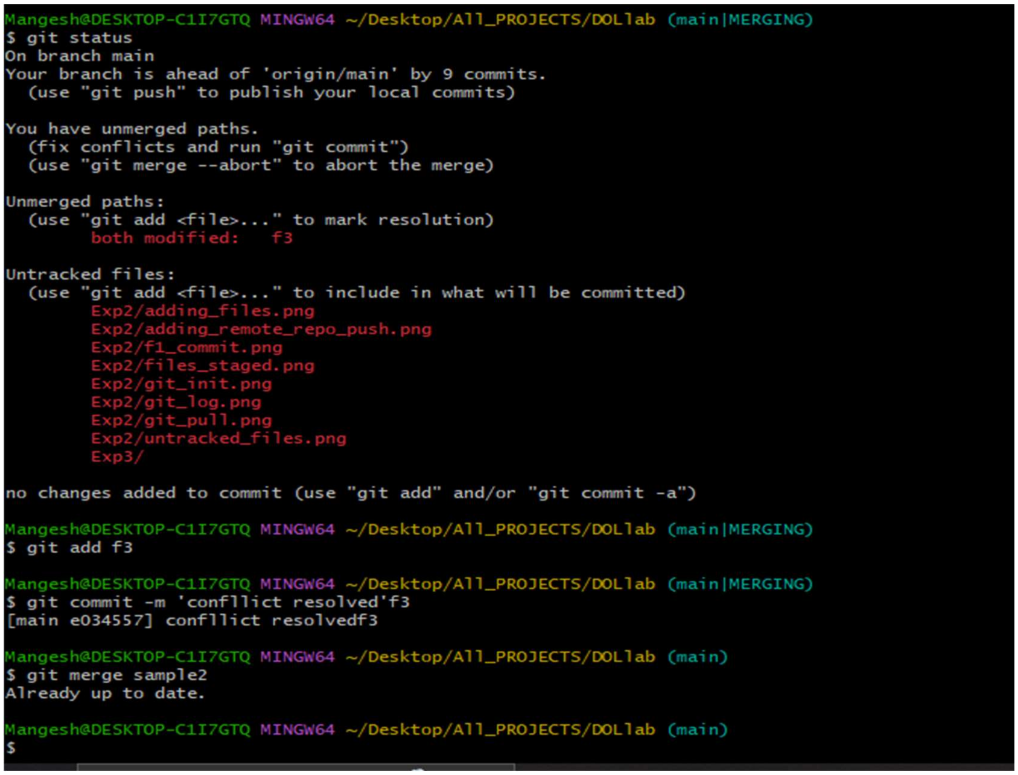
PRN: **121A3008**



```
MINGW64:/c/Users/Mangesh/Desktop/All_PROJECTS/DOLlab
Mangesh@DESKTOP-C1I7GTQ MINGW64 ~/Desktop/All_PROJECTS/DOLlab (sample2)
$ ls
Exp2/  Exp3/  f1.txt  f2.txt  f3  file2.txt

Mangesh@DESKTOP-C1I7GTQ MINGW64 ~/Desktop/All_PROJECTS/DOLlab (sample2)
$ cat >f3
line2 (changes from sample2)


Mangesh@DESKTOP-C1I7GTQ MINGW64 ~/Desktop/All_PROJECTS/DOLlab (sample2)
$ git add f3
warning: in the working copy of 'f3', LF will be replaced by CRLF the next time Git touches it

Mangesh@DESKTOP-C1I7GTQ MINGW64 ~/Desktop/All_PROJECTS/DOLlab (sample2)
$ git commit -m 'changes madein sample2 branch'f3
[sample2 a8e2fd2] changes madein sample2 branchf3
 1 file changed, 1 insertion(+), 1 deletion(-)

Mangesh@DESKTOP-C1I7GTQ MINGW64 ~/Desktop/All_PROJECTS/DOLlab (sample2)
$ git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 8 commits.
  (use "git push" to publish your local commits)

Mangesh@DESKTOP-C1I7GTQ MINGW64 ~/Desktop/All_PROJECTS/DOLlab (main)
$ cat f3
line1 (changesin main)

Mangesh@DESKTOP-C1I7GTQ MINGW64 ~/Desktop/All_PROJECTS/DOLlab (main)
$ cat >> f3
line3 ( another change from sample2)


Mangesh@DESKTOP-C1I7GTQ MINGW64 ~/Desktop/All_PROJECTS/DOLlab (main)
$ git add f3
warning: in the working copy of 'f3', LF will be replaced by CRLF the next time Git touches it

Mangesh@DESKTOP-C1I7GTQ MINGW64 ~/Desktop/All_PROJECTS/DOLlab (main)
$ git commit -m 'changes madein main branch'f3
[main 61ca4e5] changes madein main branchf3
 1 file changed, 1 insertion(+)

Mangesh@DESKTOP-C1I7GTQ MINGW64 ~/Desktop/All_PROJECTS/DOLlab (main)
$ git merge sample2
Auto-merging f3
CONFLICT (content): Merge conflict in f3
Automatic merge failed; fix conflicts and then commit the result.
```

Resolving merge conflicts:

PRN: **121A3008**



```
Mangesh@DESKTOP-C1I7GTQ MINGW64 ~/Desktop/All_PROJECTS/DOLlab (main|MERGING)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 9 commits.
  (use "git push" to publish your local commits)

You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Unmerged paths:
  (use "git add <file>..." to mark resolution)
        both modified:   f3

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Exp2/adding_files.png
        Exp2/adding_remote_repo_push.png
        Exp2/f1_commit.png
        Exp2/files_staged.png
        Exp2/git_init.png
        Exp2/git_log.png
        Exp2/git_pull.png
        Exp2/untracked_files.png
        Exp3/

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

Mangesh@DESKTOP-C1I7GTQ MINGW64 ~/Desktop/All_PROJECTS/DOLlab (main|MERGING)
$ git add f3

Mangesh@DESKTOP-C1I7GTQ MINGW64 ~/Desktop/All_PROJECTS/DOLlab (main|MERGING)
$ git commit -m 'confllict resolved'f3
[main e034557] confllict resolvedf3

Mangesh@DESKTOP-C1I7GTQ MINGW64 ~/Desktop/All_PROJECTS/DOLlab (main)
$ git merge sample2
Already up to date.

Mangesh@DESKTOP-C1I7GTQ MINGW64 ~/Desktop/All_PROJECTS/DOLlab (main)
$
```

Conclusion: Hence we have successfully performed branching and merging commands in Git.