SE489 DevOps Engineering

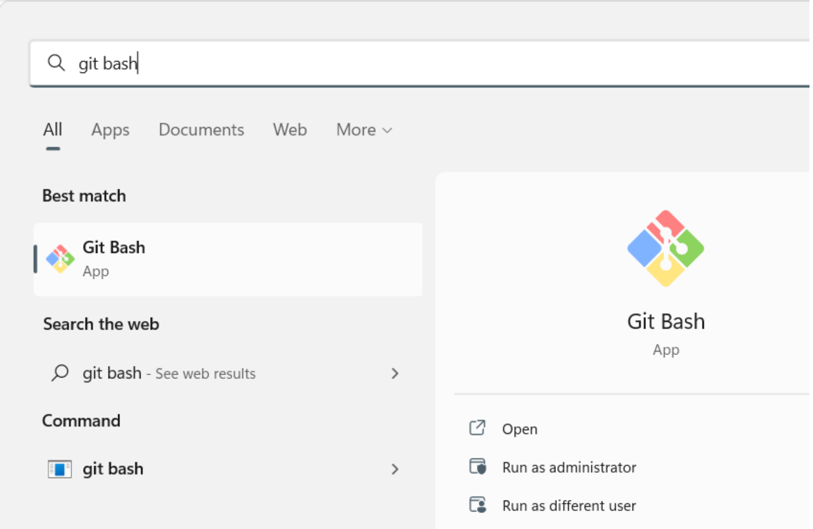
Lab 2

Lab 2: More Git Operations

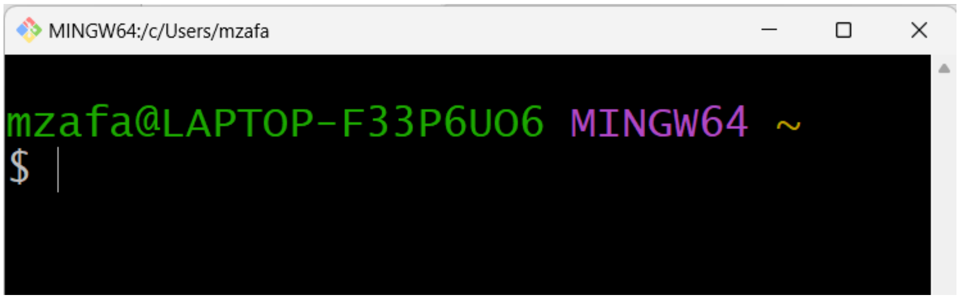
Objective: After completion of this lab session successfully, students will be able to use some of most widely used Git Operations of creating branches and merging them together.

In last lab we have used git commands from windows command prompt, now we will customize git bash shell to the local repository we have created in previous lab session.

1. From start menu, start Git by writing git bash

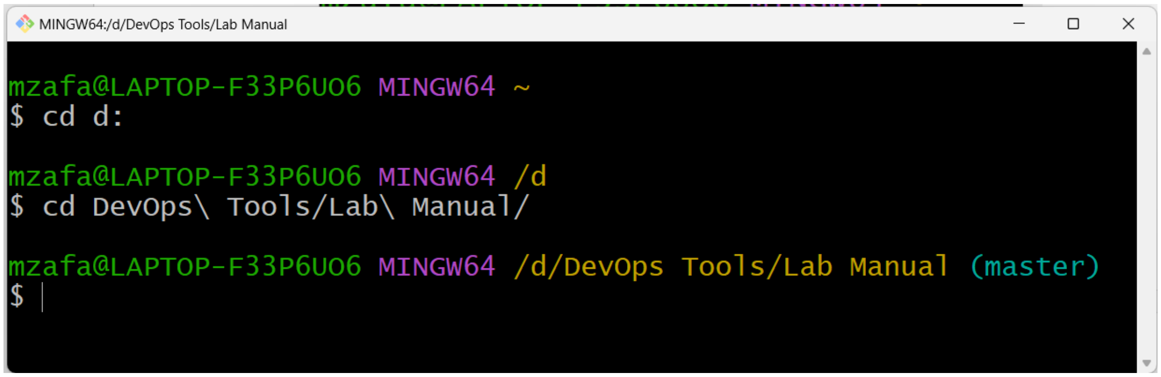


2. Git bash shell will appear, showing default directory of the installation



```
mzafa@LAPTOP-F33P6UO6 MINGW64 ~
$ 
```

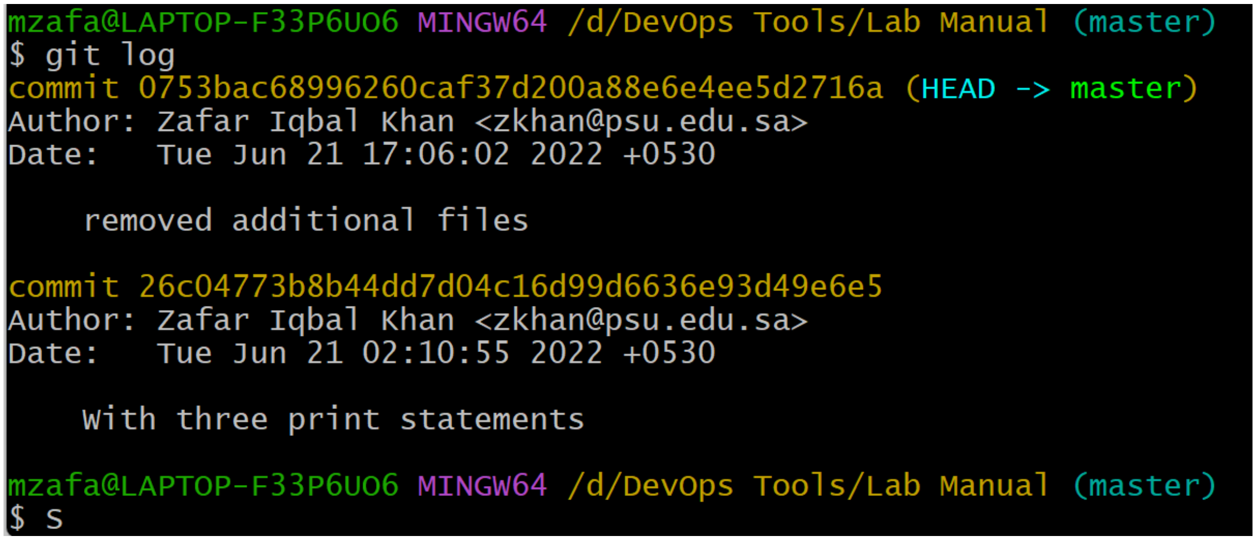
3. Git follows DOS commands, use dos commands to change the working directory to the local repository we have created at d:\DevOps Tools\Lab Manual
(Hint: because folder name consist of spaces, we can’t write them directly, use tab after writing part of the name before the space e.g. for *DevOps Tools*, write DevOps and press tab, ***DevOps*\ *Tools*** will appear)



```
mzafa@LAPTOP-F33P6U06 MINGW64 ~
$ cd d:

mzafa@LAPTOP-F33P6U06 MINGW64 /d
$ cd DevOps\ Tools/Lab\ Manual/

mzafa@LAPTOP-F33P6U06 MINGW64 /d/DevOps Tools/Lab Manual (master)
$ |
```

master at the end of the path confirms that we are now at the original branch known as **master** in the terminology of Git

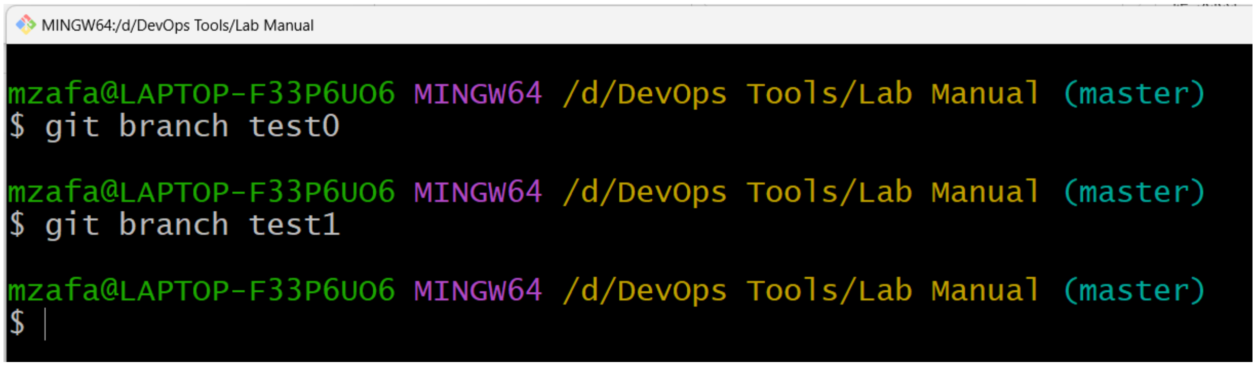
4. Before proceeding further, if we want to check the log,



```
mzafa@LAPTOP-F33P6U06 MINGW64 /d/DevOps Tools/Lab Manual (master)
$ git log
commit 0753bac68996260caf37d200a88e6e4ee5d2716a (HEAD -> master)
Author: Zafar Iqbal Khan <zkhan@psu.edu.sa>
Date:   Tue Jun 21 17:06:02 2022 +0530

    removed additional files

commit 26c04773b8b44dd7d04c16d99d6636e93d49e6e5
Author: Zafar Iqbal Khan <zkhan@psu.edu.sa>
Date:   Tue Jun 21 02:10:55 2022 +0530

    With three print statements

mzafa@LAPTOP-F33P6U06 MINGW64 /d/DevOps Tools/Lab Manual (master)
$ s
```

The string shown after commit are sha keys allotted to the commit operations, author shows who has made these commit operations, last strings are messages used with the commit operations, now it helps us to identify the commit operations.

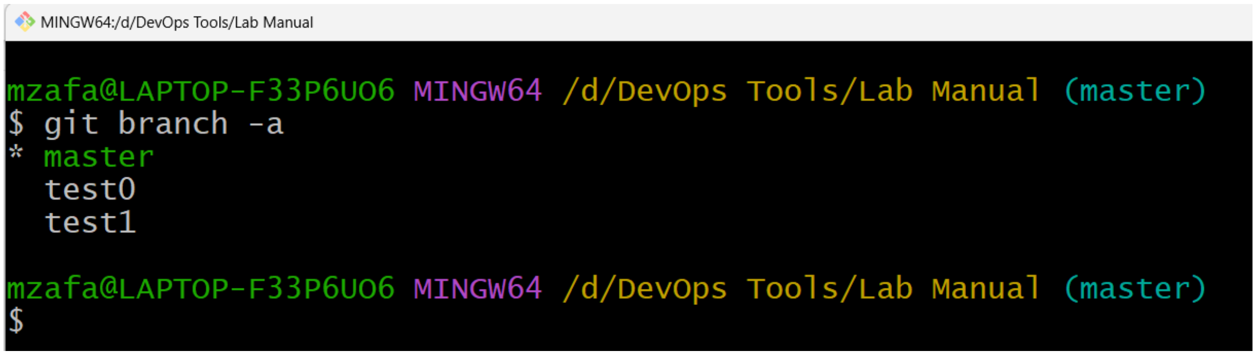
5. Now if we want to experiment with some new features, while keeping original intact, we create branches, a branch inherits all the files from the original branch ("master" in our case) Let's create two branches with the help of git branch command



```
mzafa@LAPTOP-F33P6UO6 MINGW64 /d/DevOps Tools/Lab Manual (master)
$ git branch test0

mzafa@LAPTOP-F33P6UO6 MINGW64 /d/DevOps Tools/Lab Manual (master)
$ git branch test1

mzafa@LAPTOP-F33P6UO6 MINGW64 /d/DevOps Tools/Lab Manual (master)
$ 
```

We have created two branches viz. test0 and test1

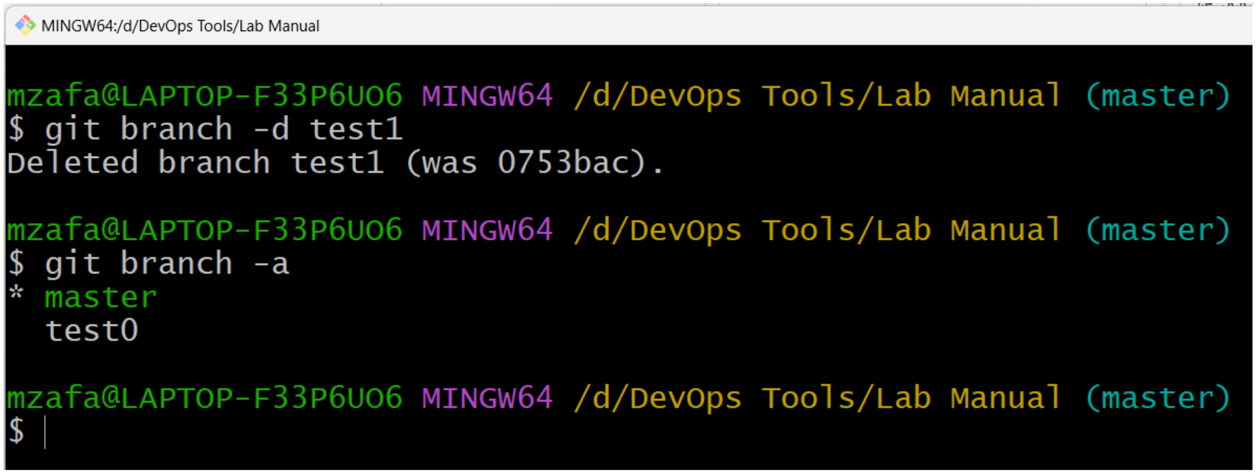
6. To list all the local as well as remote branches, -a switch is used



```
mzafa@LAPTOP-F33P6UO6 MINGW64 /d/DevOps Tools/Lab Manual (master)
$ git branch -a
* master
  test0
  test1

mzafa@LAPTOP-F33P6UO6 MINGW64 /d/DevOps Tools/Lab Manual (master)
$
```

*shows, we are at master now

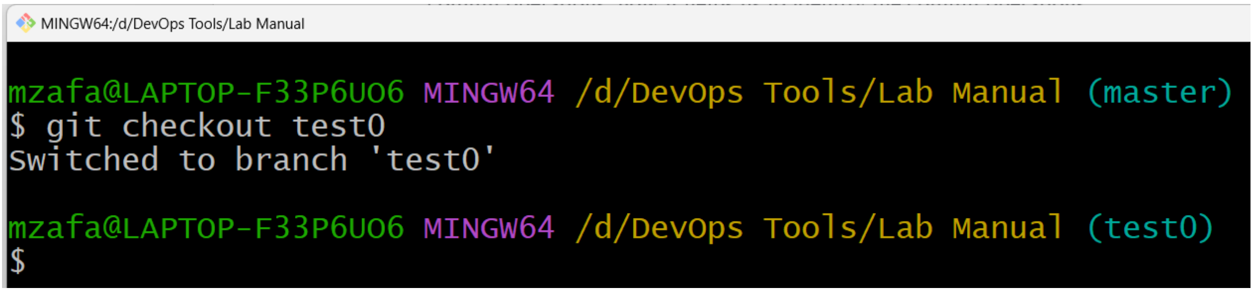
7. To delete a branch, -d switch is used, delete branch test1 and then verify it through listing



```
mzafa@LAPTOP-F33P6UO6 MINGW64 /d/DevOps Tools/Lab Manual (master)
$ git branch -d test1
Deleted branch test1 (was 0753bac).

mzafa@LAPTOP-F33P6UO6 MINGW64 /d/DevOps Tools/Lab Manual (master)
$ git branch -a
* master
  test0

mzafa@LAPTOP-F33P6UO6 MINGW64 /d/DevOps Tools/Lab Manual (master)
$ 
```

now it is showing only one branch, obviously another one has been deleted.

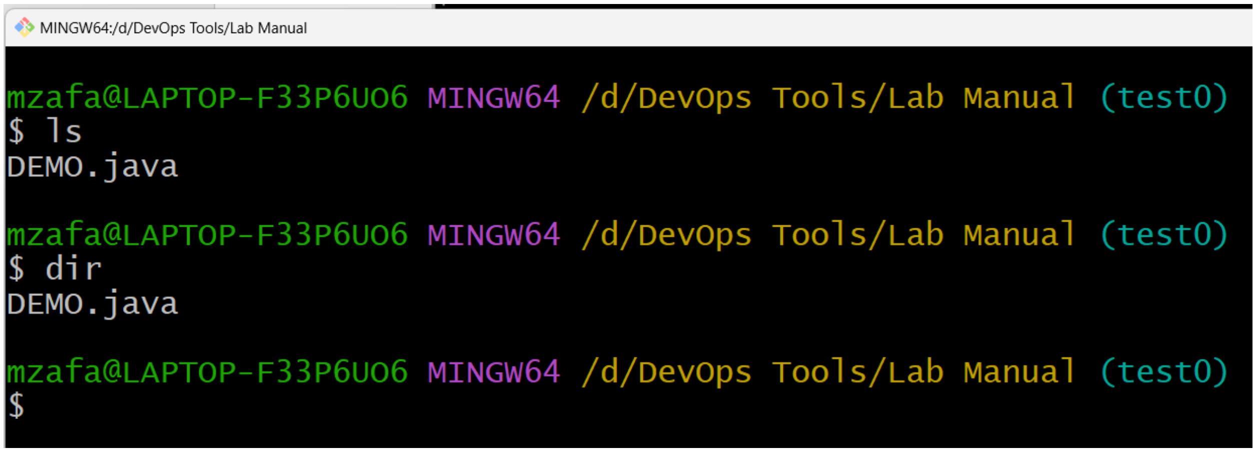
8. Now let’s switch to this branch **test0**, use checkout



```
mzafa@LAPTOP-F33P6U06 MINGW64 /d/DevOps Tools/Lab Manual (master)
$ git checkout test0
Switched to branch 'test0'

mzafa@LAPTOP-F33P6U06 MINGW64 /d/DevOps Tools/Lab Manual (test0)
$
```

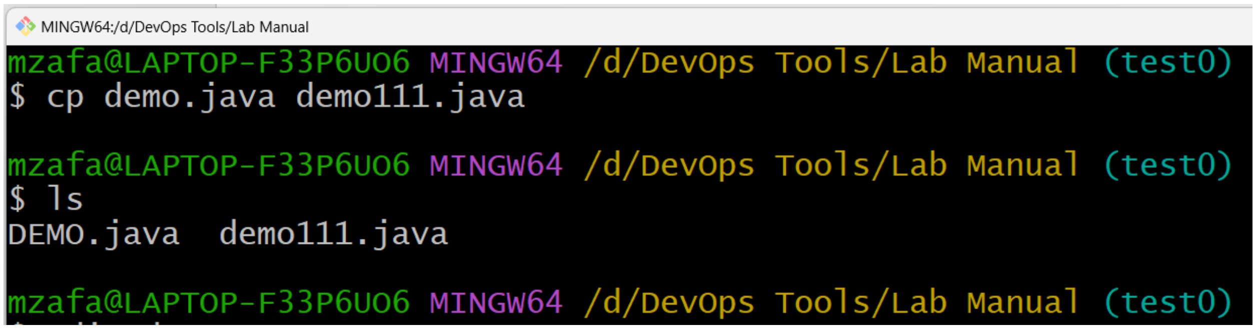
last word shows that we are in test0 branch

9. Check contents of the new branch with **ls/dir** command



```
mzafa@LAPTOP-F33P6U06 MINGW64 /d/DevOps Tools/Lab Manual (test0)
$ ls
DEMO.java

mzafa@LAPTOP-F33P6U06 MINGW64 /d/DevOps Tools/Lab Manual (test0)
$ dir
DEMO.java

mzafa@LAPTOP-F33P6U06 MINGW64 /d/DevOps Tools/Lab Manual (test0)
$
```

10. Now create a copy of demo.java file using CP command, verify it with the help of ls command

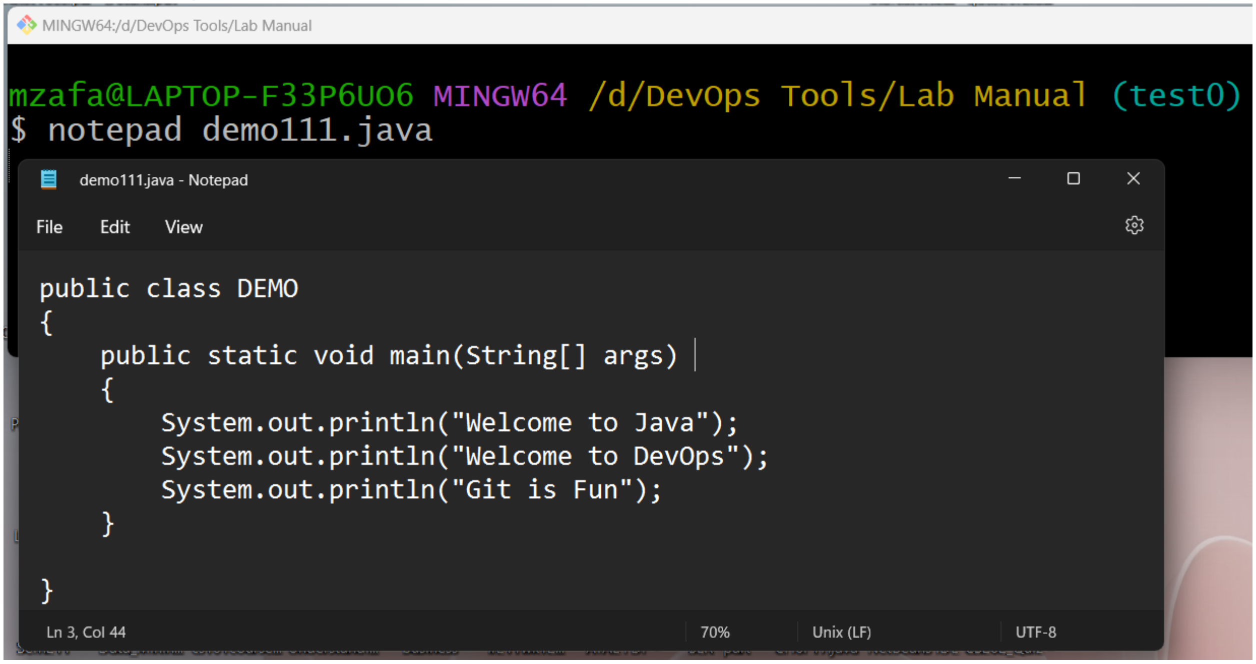


```
mzafa@LAPTOP-F33P6U06 MINGW64 /d/DevOps Tools/Lab Manual (test0)
$ cp demo.java demo111.java

mzafa@LAPTOP-F33P6U06 MINGW64 /d/DevOps Tools/Lab Manual (test0)
$ ls
DEMO.java  demo111.java

mzafa@LAPTOP-F33P6U06 MINGW64 /d/DevOps Tools/Lab Manual (test0)
```

A copy of the demo.java with name demo111.java has been created.

11. Edit this file with the default editor we have selected while installation of the git, Notepad, with following command
$notepad demo111.java, moment you will press enter, *a notepad window will open*

demo111.java file

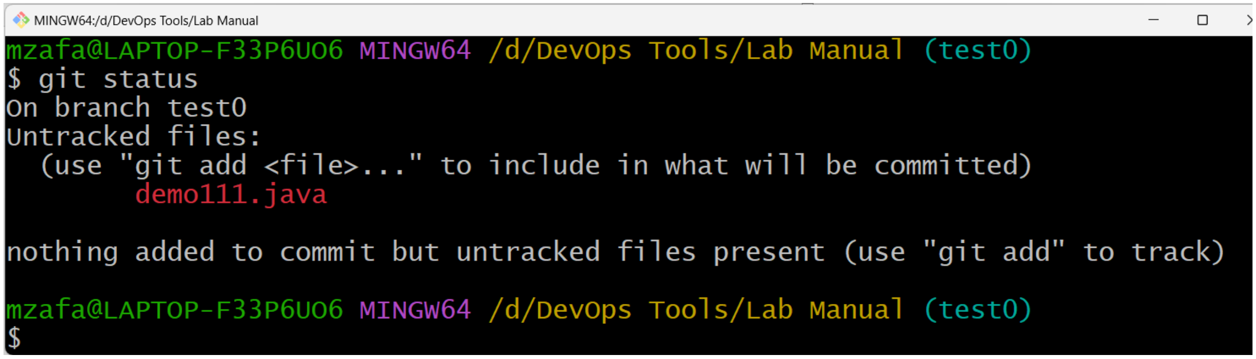


```
mzafa@LAPTOP-F33P6UO6 MINGW64 /d/DevOps Tools/Lab Manual (test0)
$ notepad demo111.java
```

```
public class DEMO
{
    public static void main(String[] args)
    {
        System.out.println("Welcome to Java");
        System.out.println("Welcome to DevOps");
        System.out.println("Git is Fun");
    }

}
```

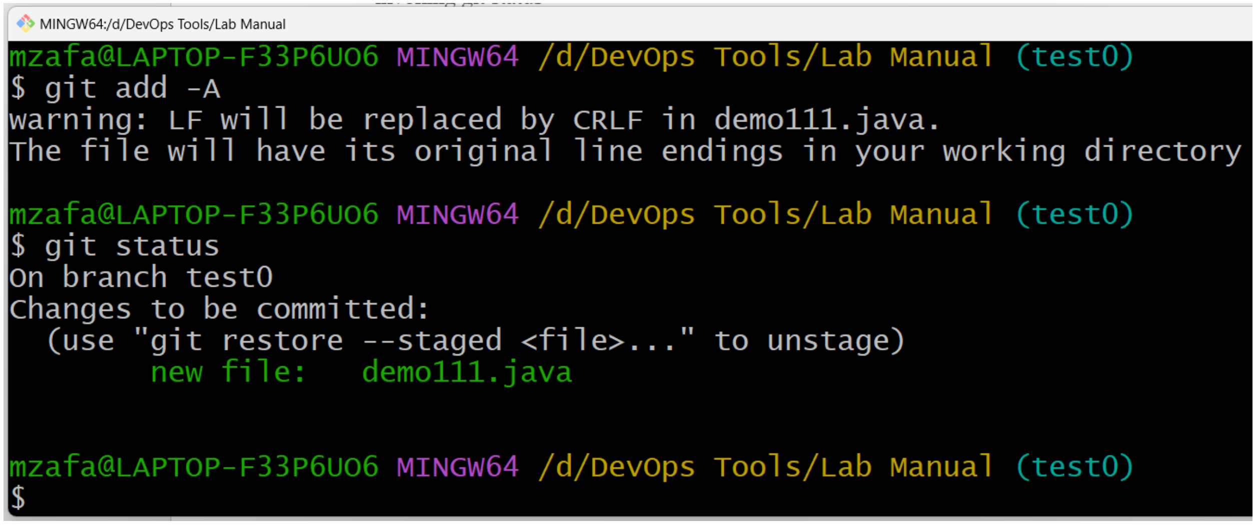
12. To make some changes, delete last print statement then save & close it.

13. Since we have created and modified a file, we can check, if it is being tracked or not by invoking git status



```
mzafa@LAPTOP-F33P6UO6 MINGW64 /d/DevOps Tools/Lab Manual (test0)
$ git status
On branch test0
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        demo111.java

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

mzafa@LAPTOP-F33P6UO6 MINGW64 /d/DevOps Tools/Lab Manual (test0)
$
```

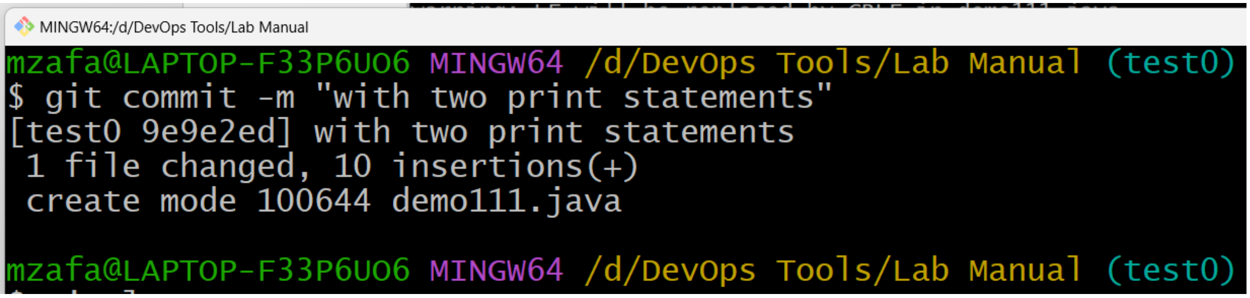
14. Now let's add this to stage area and then check the status again



```
mzafa@LAPTOP-F33P6UO6 MINGW64 /d/DevOps Tools/Lab Manual (test0)
$ git add -A
warning: LF will be replaced by CRLF in demo111.java.
The file will have its original line endings in your working directory

mzafa@LAPTOP-F33P6UO6 MINGW64 /d/DevOps Tools/Lab Manual (test0)
$ git status
On branch test0
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   demo111.java


mzafa@LAPTOP-F33P6UO6 MINGW64 /d/DevOps Tools/Lab Manual (test0)
$
```

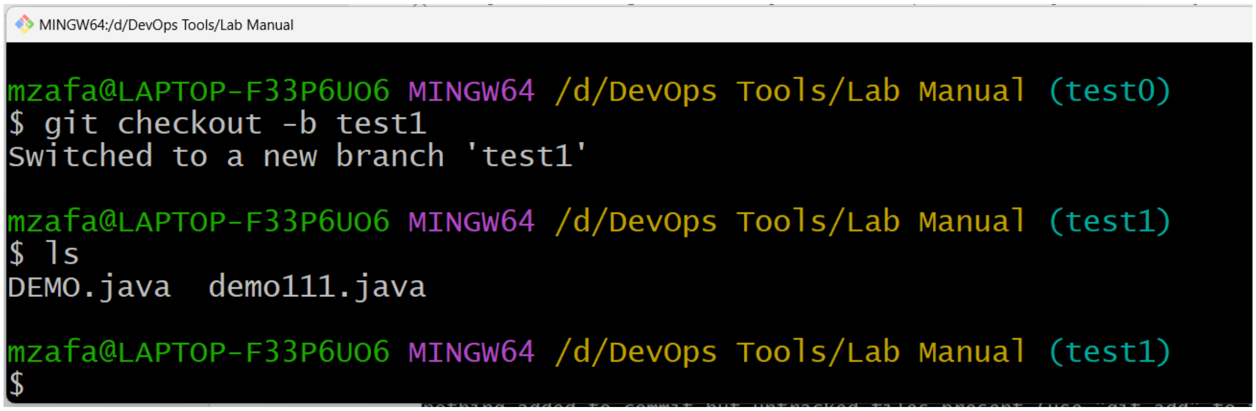
-A, when used with add, adds all tracked and tracked files to staging area (file will be tracked)

15. Commit this change with message tag, *with two print statements.*



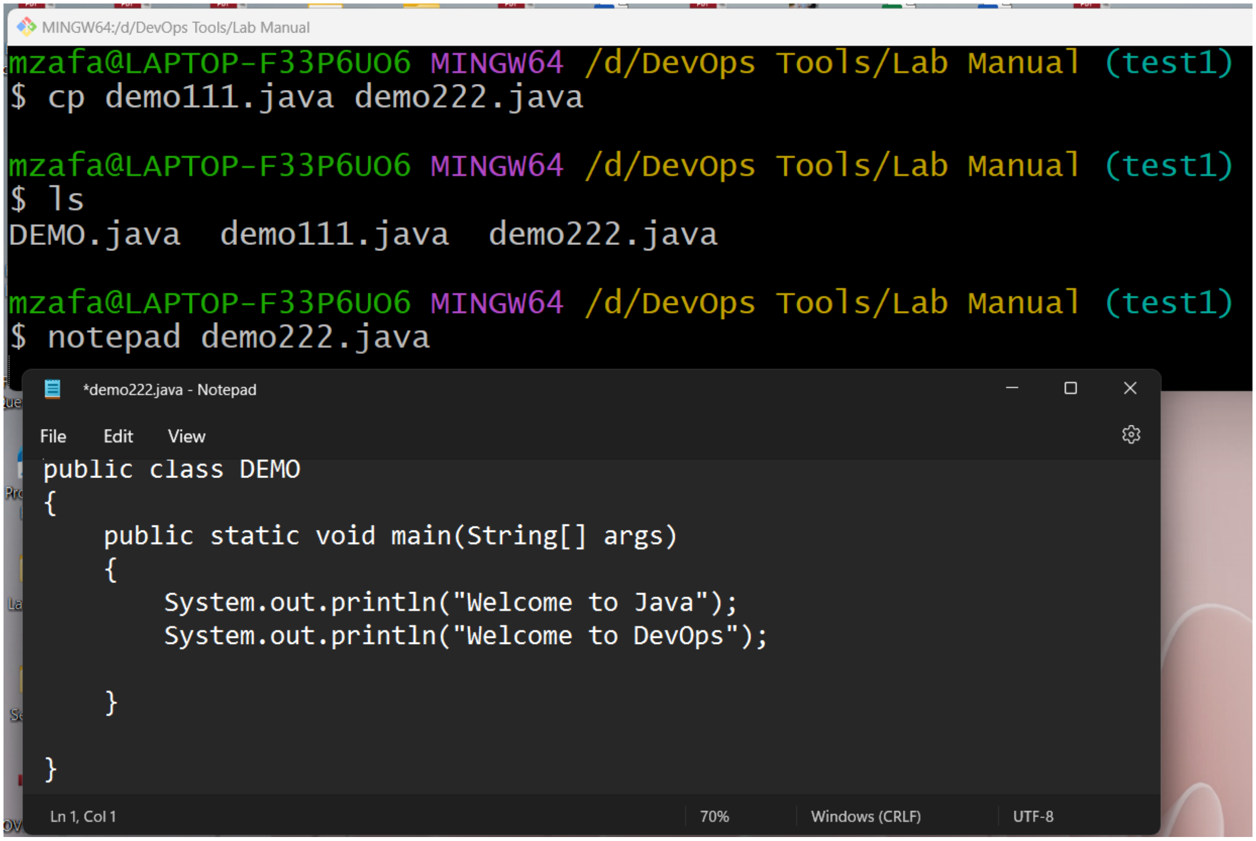
```
mzafa@LAPTOP-F33P6UO6 MINGW64 /d/DevOps Tools/Lab Manual (test0)
$ git commit -m "with two print statements"
[test0 9e9e2ed] with two print statements
 1 file changed, 10 insertions(+)
 create mode 100644 demo111.java

mzafa@LAPTOP-F33P6UO6 MINGW64 /d/DevOps Tools/Lab Manual (test0)
```

16. We can also create a branch with the help of checkout command by using switch **-b** and branch name, and check the contents with **ls**



```
mzafa@LAPTOP-F33P6UO6 MINGW64 /d/DevOps Tools/Lab Manual (test0)
$ git checkout -b test1
Switched to a new branch 'test1'

mzafa@LAPTOP-F33P6UO6 MINGW64 /d/DevOps Tools/Lab Manual (test1)
$ ls
DEMO.java  demo111.java

mzafa@LAPTOP-F33P6UO6 MINGW64 /d/DevOps Tools/Lab Manual (test1)
$
```

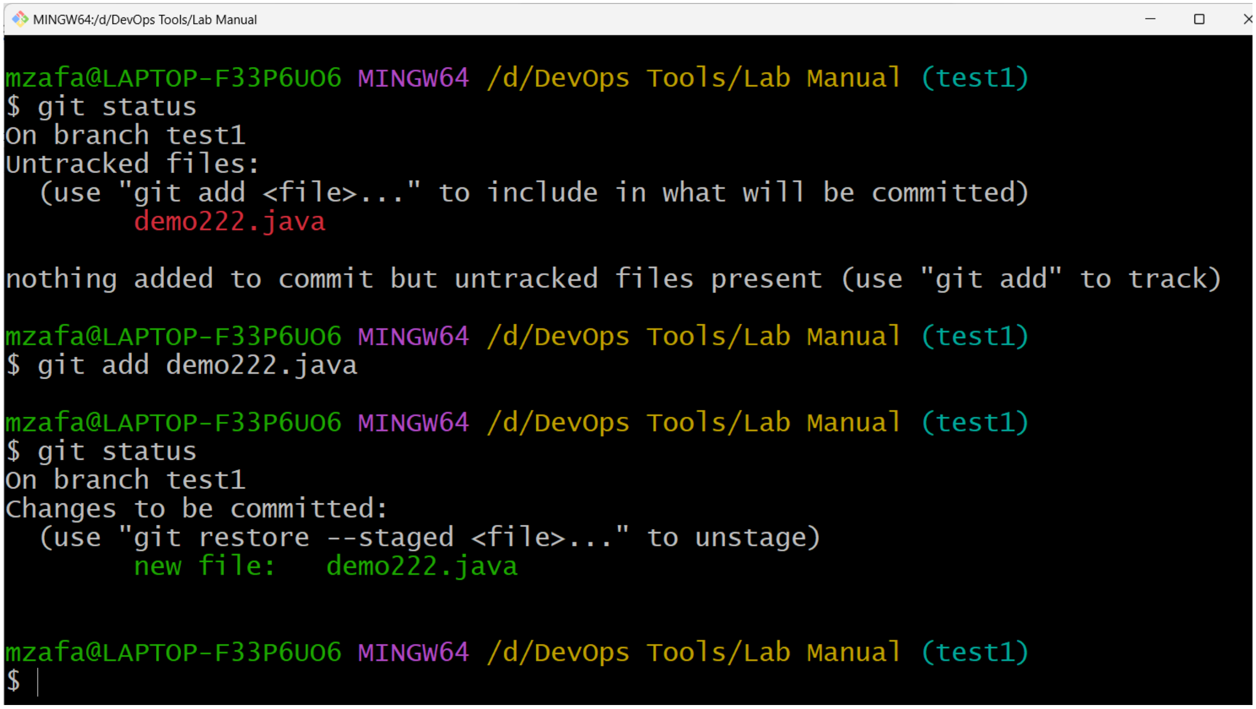
17. Repeat steps from 10 -12 to create and edit the copy, rename the file **demo222.java**, time delete the second print statement as well so as only one print statement remains in the file.



```
mzafa@LAPTOP-F33P6U06 MINGW64 /d/DevOps Tools/Lab Manual (test1)
$ cp demo111.java demo222.java

mzafa@LAPTOP-F33P6U06 MINGW64 /d/DevOps Tools/Lab Manual (test1)
$ ls
DEMO.java  demo111.java  demo222.java

mzafa@LAPTOP-F33P6U06 MINGW64 /d/DevOps Tools/Lab Manual (test1)
$ notepad demo222.java
```

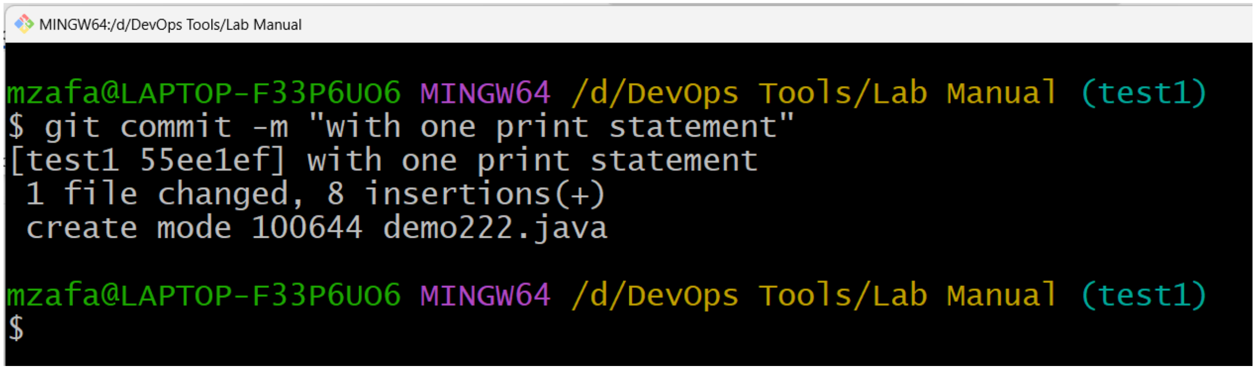
```
public class DEMO
{
    public static void main(String[] args)
    {
        System.out.println("Welcome to Java");
        System.out.println("Welcome to DevOps");

    }

}
```

18. Same as previously, lets check status, add this file and check status again.



```

mzafa@LAPTOP-F33P6U06 MINGW64 /d/DevOps Tools/Lab Manual (test1)
$ git status
On branch test1
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        demo222.java

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

mzafa@LAPTOP-F33P6U06 MINGW64 /d/DevOps Tools/Lab Manual (test1)
$ git add demo222.java

mzafa@LAPTOP-F33P6U06 MINGW64 /d/DevOps Tools/Lab Manual (test1)
$ git status
On branch test1
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   demo222.java


mzafa@LAPTOP-F33P6U06 MINGW64 /d/DevOps Tools/Lab Manual (test1)
$ |
```

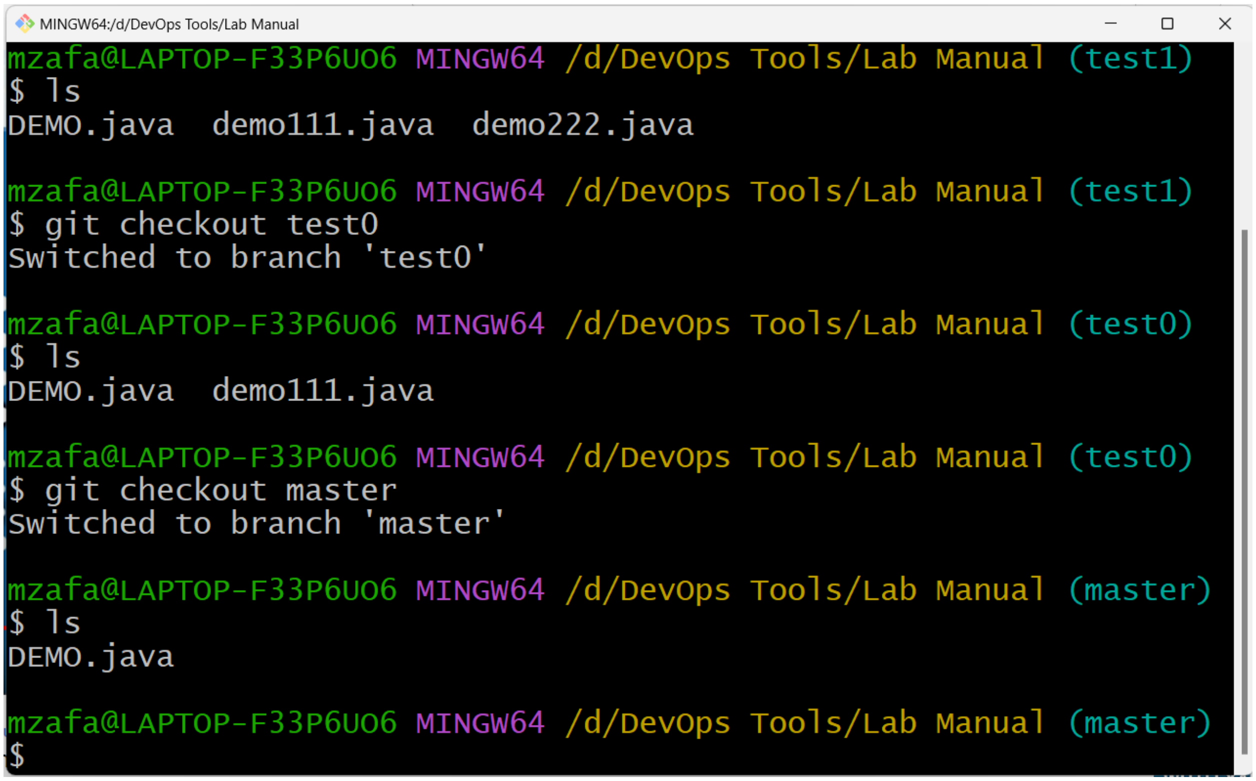
19. Commit this file with message tag, “with one print statement”.



MINGW64:/d/DevOps Tools/Lab Manual

```
mzafa@LAPTOP-F33P6UO6 MINGW64 /d/DevOps Tools/Lab Manual (test1)
$ git commit -m "with one print statement"
[test1 55ee1ef] with one print statement
 1 file changed, 8 insertions(+)
 create mode 100644 demo222.java

mzafa@LAPTOP-F33P6UO6 MINGW64 /d/DevOps Tools/Lab Manual (test1)
$
```

20. Use successive checkout and ls, to look into various branches and their contents.

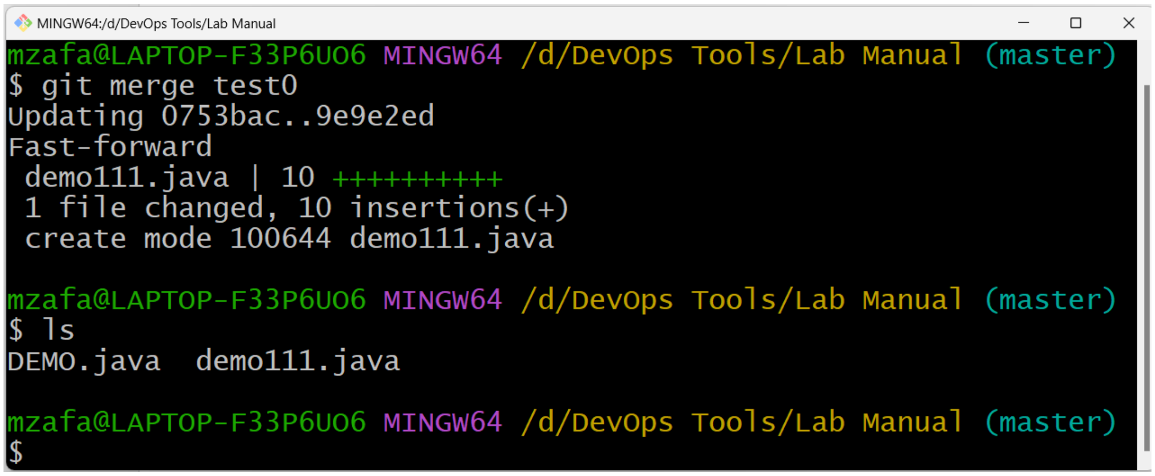


MINGW64:/d/DevOps Tools/Lab Manual

```
mzafa@LAPTOP-F33P6UO6 MINGW64 /d/DevOps Tools/Lab Manual (test1)
$ ls
DEMO.java  demo111.java  demo222.java

mzafa@LAPTOP-F33P6UO6 MINGW64 /d/DevOps Tools/Lab Manual (test1)
$ git checkout test0
Switched to branch 'test0'

mzafa@LAPTOP-F33P6UO6 MINGW64 /d/DevOps Tools/Lab Manual (test0)
$ ls
DEMO.java  demo111.java

mzafa@LAPTOP-F33P6UO6 MINGW64 /d/DevOps Tools/Lab Manual (test0)
$ git checkout master
Switched to branch 'master'

mzafa@LAPTOP-F33P6UO6 MINGW64 /d/DevOps Tools/Lab Manual (master)
$ ls
DEMO.java

mzafa@LAPTOP-F33P6UO6 MINGW64 /d/DevOps Tools/Lab Manual (master)
$
```

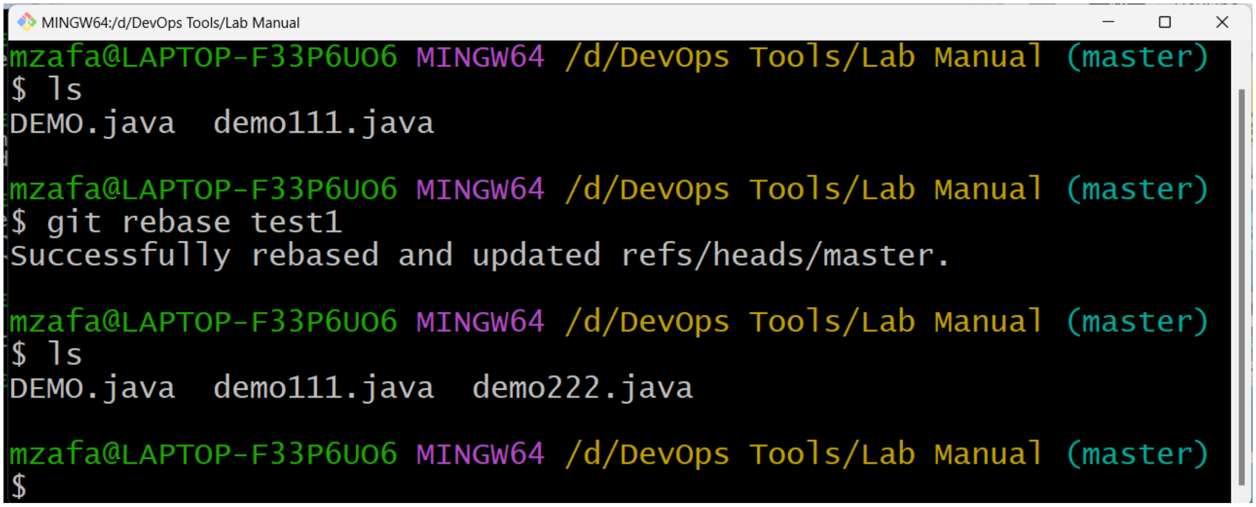
21. We can use **merge** to merge files from various branches, point to note that, we must be at the destination branch of the merge operation.

being at master branch, we will merge contents from branch test0 into branch master



```
mzafa@LAPTOP-F33P6UO6 MINGW64 /d/DevOps Tools/Lab Manual (master)
$ git merge test0
Updating 0753bac..9e9e2ed
Fast-forward
 demo111.java | 10 ++++++++++
 1 file changed, 10 insertions(+)
 create mode 100644 demo111.java

mzafa@LAPTOP-F33P6UO6 MINGW64 /d/DevOps Tools/Lab Manual (master)
$ ls
DEMO.java  demo111.java

mzafa@LAPTOP-F33P6UO6 MINGW64 /d/DevOps Tools/Lab Manual (master)
$
```

22. Rebase command is used to rebase the project, rebasing merges all files into one and produces much clear structure of project tree.



```
mzafa@LAPTOP-F33P6UO6 MINGW64 /d/DevOps Tools/Lab Manual (master)
$ ls
DEMO.java  demo111.java

mzafa@LAPTOP-F33P6UO6 MINGW64 /d/DevOps Tools/Lab Manual (master)
$ git rebase test1
Successfully rebased and updated refs/heads/master.

mzafa@LAPTOP-F33P6UO6 MINGW64 /d/DevOps Tools/Lab Manual (master)
$ ls
DEMO.java  demo111.java  demo222.java

mzafa@LAPTOP-F33P6UO6 MINGW64 /d/DevOps Tools/Lab Manual (master)
$
```

As obvious from the output produced, all files from the branch test1 have been merged into master.