

# Java full stack task-1

Name : Yogeshwaran D

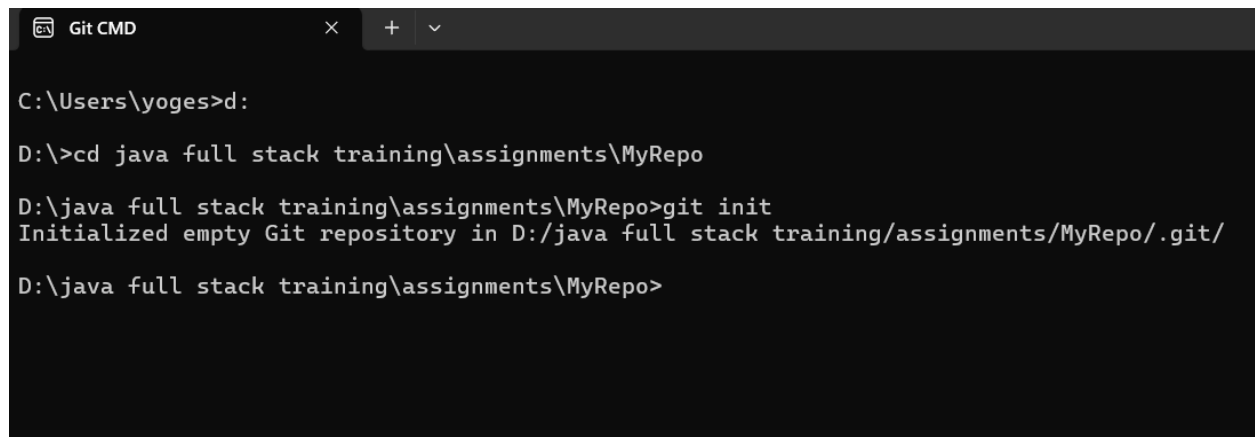
KSRIET – CSE FINAL YEAR

DATE : 11-07-2024

Git Lab assignment-1 :

---

1. Create a local git repo with the name "MyRepo".

A screenshot of a terminal window titled "Git CMD". The terminal shows the following commands and output:

```
C:\Users\yoges>d:
D:\>cd java full stack training\assignments\MyRepo
D:\java full stack training\assignments\MyRepo>git init
Initialized empty Git repository in D:/java full stack training/assignments/MyRepo/.git/
D:\java full stack training\assignments\MyRepo>
```

- ❖ Empty git repository has been initialized.
- ❖ 2. create one .txt file, five .c files and five .java files in the working area.

MyRepo

MyRepo

+

←

→

↑

↻

🖨

>

This PC

>

Data (D:)

>

java full stack training

>

assignments

>

MyRepo

+

New ▾

✂

📄

📁

🔍

🔗

🗑

↕

Sort ▾

≡

View ▾

⋮

🏠

Home

🖼

Gallery

>

📁

OneDrive - Persona

↓

Downloads

🎵

Music

📁

house of dragons s

📁

season 2

📁

repo

📁

MyRepo

### 3. Verify the git status

```
D:\java full stack training\assignments\MyRepo>git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    file1.txt
    file2.c
    file3.c
    file4.c
    file5.c
    file6.c
    file7.java
    file8.java

nothing added to commit but untracked files present (use "git add" to track)
D:\java full stack training\assignments\MyRepo>|
```

### 4. Add only the .txt into the staging area

```
nothing added to commit but untracked files present (use "git add" to track)
D:\java full stack training\assignments\MyRepo>git add file1.txt
D:\java full stack training\assignments\MyRepo>|
```

## 5. verify the git status

```
D:\java full stack training\assignments\MyRepo>git add file1.txt
D:\java full stack training\assignments\MyRepo>git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   file1.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        file2.c
        file3.c
        file4.c
        file5.c
        file6.c
        file7.java
        file8.java

D:\java full stack training\assignments\MyRepo>|
```

## 6. Make a commit with the name "TXT Commit"

```
D:\java full stack training\assignments\MyRepo>git commit -m "TXT commit"
[master (root-commit) b0579f0] TXT commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 file1.txt

D:\java full stack training\assignments\MyRepo>
```

## 7. Add all .c files into the staging area

```
D:\java full stack training\assignments\MyRepo>git add *.c

D:\java full stack training\assignments\MyRepo>
```

## 8. verify the git status

```
D:\java full stack training\assignments\MyRepo>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   file2.c
        new file:   file3.c
        new file:   file4.c
        new file:   file5.c
        new file:   file6.c

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        file7.java
        file8.java

D:\java full stack training\assignments\MyRepo>
```

## 9. Make a commit with the name "C Commit"

```
D:\java full stack training\assignments\MyRepo>git commit -m "C commit"
[master 6c57c7e] C commit
 5 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 file2.c
 create mode 100644 file3.c
 create mode 100644 file4.c
 create mode 100644 file5.c
 create mode 100644 file6.c

D:\java full stack training\assignments\MyRepo>|
```

## 10 .Add all .java files into the staging area

```
D:\java full stack training\assignments\MyRepo>git add *.java

D:\java full stack training\assignments\MyRepo>|
```

## 11. verify the git status

```
D:\java full stack training\assignments\MyRepo>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   file7.java
        new file:   file8.java

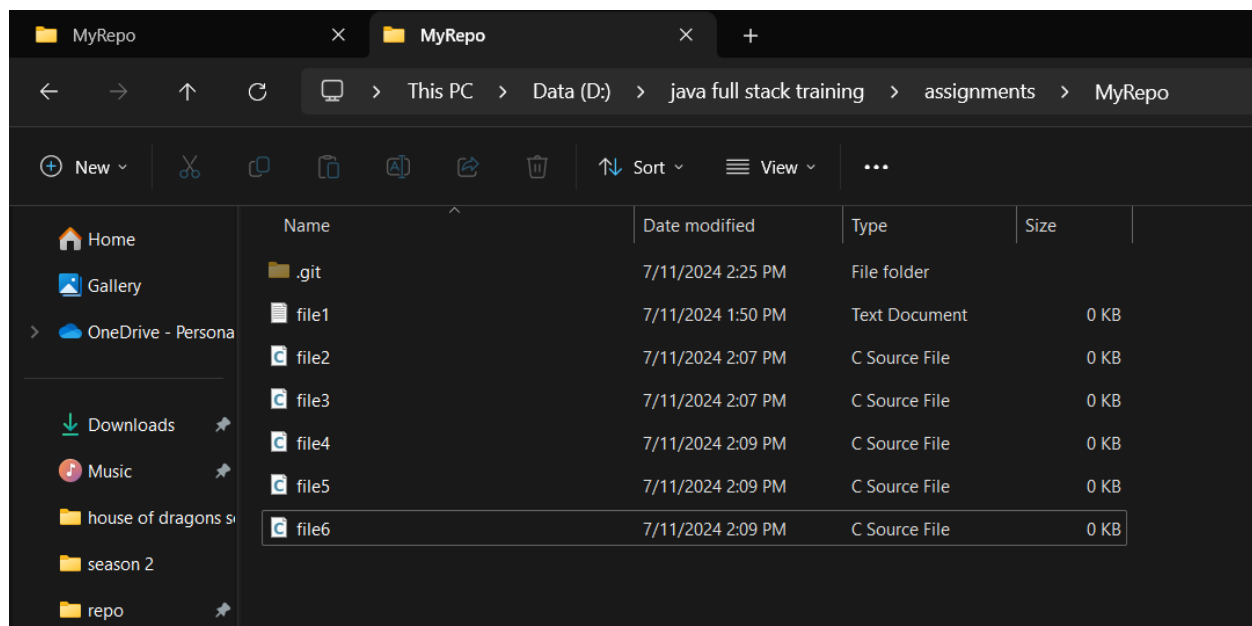
D:\java full stack training\assignments\MyRepo>|
```

## 12. Make a commit with the name "Java Commit"

```
D:\java full stack training\assignments\MyRepo>git commit -m "JAVA commit"
[master bf2c5e7] JAVA commit
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 file7.java
 create mode 100644 file8.java

D:\java full stack training\assignments\MyRepo>
```

### 13. Delete all .java files from working area



### 14. Now see the status

```
D:\java full stack training\assignments\MyRepo>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:    file7.java
        deleted:    file8.java

no changes added to commit (use "git add" and/or "git commit -a")
D:\java full stack training\assignments\MyRepo>|
```

15. add the changes into the staging area

```
D:\java full stack training\assignments\MyRepo>git add .
```

16. Make a commit with the name "Delete Java Commit"

```
D:\java full stack training\assignments\MyRepo>git commit -m "JAVA delete commit"
[master aa19421] JAVA delete commit
 2 files changed, 0 insertions(+), 0 deletions(-)
 delete mode 100644 file7.java
 delete mode 100644 file8.java

D:\java full stack training\assignments\MyRepo>|
```

17. display all commit details



---

```
D:\java full stack training\assignments\MyRepo>git log
commit aa19421620f9a246c577192a69cd88473f95de73 (HEAD -> master)
Author: Yogeshwaran <yogeshjithashni@gmail.com>
Date: Thu Jul 11 14:31:22 2024 +0530
```

JAVA delete commit

```
commit bf2c5e7caa92fe33fe415ffa0d324bbe0ace2fe3
Author: Yogeshwaran <yogeshjithashni@gmail.com>
Date: Thu Jul 11 14:25:26 2024 +0530
```

JAVA commit

```
commit 6c57c7eebfbf1592349973ed40a359b450121bdb
Author: Yogeshwaran <yogeshjithashni@gmail.com>
Date: Thu Jul 11 14:22:08 2024 +0530
```

C commit

```
commit b0579f04ba5bebc2b15d2c28ea95184e03edac52
Author: Yogeshwaran <yogeshjithashni@gmail.com>
Date: Thu Jul 11 14:18:27 2024 +0530
```

TXT commit

---

## DAY – 2:

---

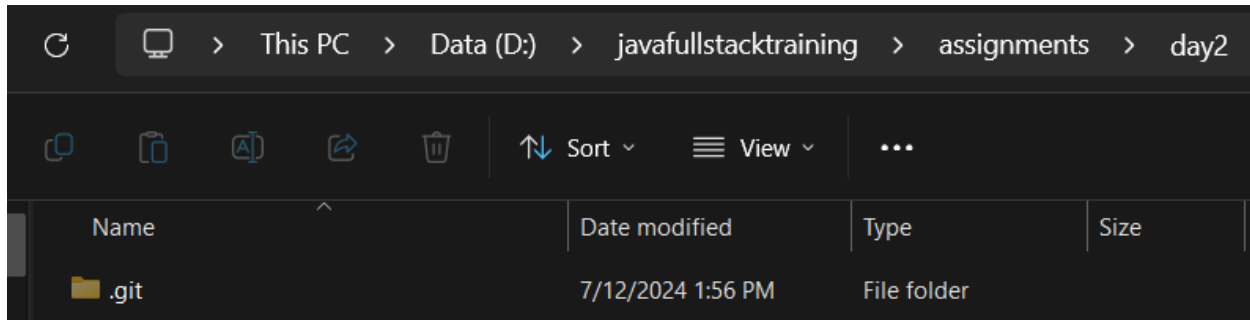
1. Create a new directory and change into it.

```
C:\Users\yoges>cd ..
C:\Users>cd ..
C:\>d:
D:\>|
```

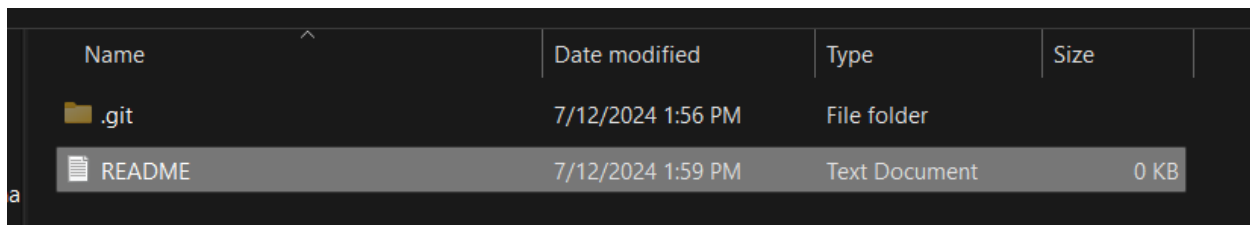
2. Use the init command to create a Git repository in that directory.

```
C:\Users\yoges>cd..  
C:\Users>cd..  
C:\>d:  
D:\>cd javafullstacktraining\assignments\day2  
D:\javafullstacktraining\assignments\day2>git init  
Initialized empty Git repository in D:/javafullstacktraining/assignments/day2/.git/  
D:\javafullstacktraining\assignments\day2>|
```

3. Observe that there is now a .git directory.



4. Create a README file.



5. Look at the output of the status command; the README you created should appear as an untracked file.

```
D:\javafullstacktraining\assignments\day2>git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        README.txt

nothing added to commit but untracked files present (use "git add" to track)
D:\javafullstacktraining\assignments\day2>
```

6. Use the add command to add the new file to the staging area. Again, look at the output of the status command.

```
D:\javafullstacktraining\assignments\day2>git add README.txt

D:\javafullstacktraining\assignments\day2>git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   README.txt

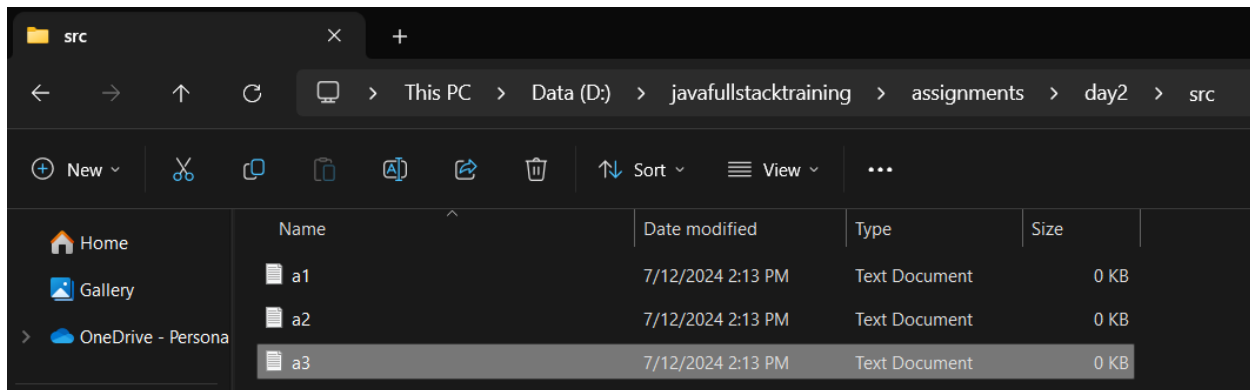
D:\javafullstacktraining\assignments\day2>
```

7. Now use the commit command to commit the contents of the staging area.

```
D:\javafullstacktraining\assignments\day2>git commit -m "adding a README file"
[master (root-commit) 76a9ca7] adding a README file
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 README.txt

D:\javafullstacktraining\assignments\day2>
```

8. Create a src directory and add a couple of files to it.



9. Use the add command, but name the directory, not the individual files. Use the status command. See how both files have been staged. Commit them.

```
D:\javafullstacktraining\assignments\day2>git add src

D:\javafullstacktraining\assignments\day2>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   src/a1.txt
        new file:   src/a2.txt
        new file:   src/a3.txt

D:\javafullstacktraining\assignments\day2>git commit -m "adding a src"
[master 52da11f] adding a src
 3 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 src/a1.txt
 create mode 100644 src/a2.txt
 create mode 100644 src/a3.txt
```

10. Make a change to one of the files. Use the diff command to view the details of the change.

```
D:\javafullstacktraining\assignments\day2>git diff 52da11f98faa0681c31680b37f6ee930d5a84e0f 76a9ca7de0488f155353dbddcbeb30f93e6b63fe
diff --git a/src/a1.txt b/src/a1.txt
deleted file mode 100644
index e69de29..0000000
diff --git a/src/a2.txt b/src/a2.txt
deleted file mode 100644
index e69de29..0000000
diff --git a/src/a3.txt b/src/a3.txt
deleted file mode 100644
index e69de29..0000000

D:\javafullstacktraining\assignments\day2>
```

11. Next, add the changed file, and notice how it moves to the staging area in the status output. Also observe that the diff command you did before using add now gives no output. Why not? What do you have to do to see a diff of the things in the staging area? (Hint: review the slides if you can't remember.)

```
D:\javafullstacktraining\assignments\day2>git diff
diff --git a/src/a2.txt b/src/a2.txt
deleted file mode 100644
index e69de29..0000000
diff --git a/src/a3.txt b/src/a3.txt
index e69de29..5969a5c 100644
--- a/src/a3.txt
+++ b/src/a3.txt
@@ -0,0 +1 @@
+yogesh
\ No newline at end of file

D:\javafullstacktraining\assignments\day2>git add .

D:\javafullstacktraining\assignments\day2>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        deleted:      src/a2.txt
        modified:     src/a3.txt

D:\javafullstacktraining\assignments\day2>|
```

12. Now – without committing – make another change to the same file you changed in step 10. Look at the status

output, and the diff output. Notice how you can have both staged and unstaged changes, even when you're talking about a single file. Observe the difference when you use the add command to stage the latest round of changes. Finally, commit them. You should now have started to get a feel for the staging area.

```
D:\javafullstacktraining\assignments\day2>git diff
diff --git a/src/a3.txt b/src/a3.txt
index 5969a5c..c3a6ac6 100644
--- a/src/a3.txt
+++ b/src/a3.txt
@@ -1,1 @@
-yogesh
\ No newline at end of file
+yog
\ No newline at end of file

D:\javafullstacktraining\assignments\day2>
```

13. Use the log command in order to see all of the commits you made so far.

```
D:\javafullstacktraining\assignments\day2>git add .

D:\javafullstacktraining\assignments\day2>git diff

D:\javafullstacktraining\assignments\day2>git commit -m "msg"
[master aa82aeb] msg
 2 files changed, 1 insertion(+)
 delete mode 100644 src/a2.txt

D:\javafullstacktraining\assignments\day2>git log
commit aa82aebb7099a64a495b3a96c537e3024d19784f (HEAD -> master)
Author: Yogeshwaran <yogeshjithashni@gmail.com>
Date:   Fri Jul 12 15:06:43 2024 +0530

    msg

commit 52da11f98faa0681c31680b37f6ee930d5a84e0f
Author: Yogeshwaran <yogeshjithashni@gmail.com>
Date:   Fri Jul 12 14:19:54 2024 +0530

    adding a src

commit 76a9ca7de0488f155353dbddcb30f93e6b63fe
Author: Yogeshwaran <yogeshjithashni@gmail.com>
Date:   Fri Jul 12 14:02:52 2024 +0530

    adding a README file

D:\javafullstacktraining\assignments\day2>
```

14. Use the show command to look at an individual commit. How many characters of the commit identifier can you get away with typing at a minimum?

```
D:\javafullstacktraining\assignments\day2>git show
commit aa82aebb7099a64a495b3a96c537e3024d19784f (HEAD -> master)
Author: Yogeshwaran <yogeshjithashni@gmail.com>
Date:   Fri Jul 12 15:06:43 2024 +0530

    msg

diff --git a/src/a2.txt b/src/a2.txt
deleted file mode 100644
index e69de29..0000000
diff --git a/src/a3.txt b/src/a3.txt
index e69de29..c3a6ac6 100644
--- a/src/a3.txt
+++ b/src/a3.txt
@@ -0,0 +1 @@
+yog
\ No newline at end of file

D:\javafullstacktraining\assignments\day2>
```

15. Make a couple more commits, at least one of which should add an extra file



```
D:\javafullstacktraining\assignments\day2>git add .

D:\javafullstacktraining\assignments\day2>git commit -m "added a file"
[master a4239cd] added a file
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 src/a2.txt

D:\javafullstacktraining\assignments\day2>git log
commit a4239cd35fd30192b2e8899f6a2a036018fa935c (HEAD -> master)
Author: Yogeshwaran <yogeshjithashni@gmail.com>
Date: Fri Jul 12 15:16:16 2024 +0530

    added a file

commit aa82aebb7099a64a495b3a96c537e3024d19784f
Author: Yogeshwaran <yogeshjithashni@gmail.com>
Date: Fri Jul 12 15:06:43 2024 +0530

    msg

commit 52da11f98faa0681c31680b37f6ee930d5a84e0f
Author: Yogeshwaran <yogeshjithashni@gmail.com>
Date: Fri Jul 12 14:19:54 2024 +0530

    adding a src

commit 76a9ca7de0488f155353dbddcbeb30f93e6b63fe
Author: Yogeshwaran <yogeshjithashni@gmail.com>
Date: Fri Jul 12 14:02:52 2024 +0530
```

---

## Exercise 2

---

1. Run the status command.

Notice how it tells you what branch you are in.

```

C:\Users\Admin>e:
E:\>cd FULLSTACK
E:\FULLSTACK>cd Exercise2
E:\FULLSTACK\Exercise2>git init
Initialized empty Git repository in E:/FULLSTACK/Exercise2/.git/
E:\FULLSTACK\Exercise2>git add .
E:\FULLSTACK\Exercise2>git commit -m "ex2"
[master (root-commit) 626bae1] ex2
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 b1.txt
 create mode 100644 b2.txt
E:\FULLSTACK\Exercise2>cd ..
E:\FULLSTACK>cd CloneEx2
E:\FULLSTACK\CloneEx2>git clone E:\FULLSTACK\Exercise2
Cloning into 'Exercise2'...
done.
E:\FULLSTACK\CloneEx2>git clone E:\FULLSTACK\Exercise2 .
fatal: destination path '.' already exists and is not an empty directory.
E:\FULLSTACK\CloneEx2>cd ..
E:\FULLSTACK>cd CloneEx2
E:\FULLSTACK\CloneEx2>git clone E:\FULLSTACK\Exercise2 .
Cloning into '.'...
done.
E:\FULLSTACK\CloneEx2>

```

2. Use the branch command to create a new branch.

```

E:\FULLSTACK\CloneEx2>git branch JB
E:\FULLSTACK\CloneEx2>git branch
JB
* master

```

3. Use the checkout command to switch to it.

```
E:\FULLSTACK\CloneEx2>git checkout JB
Switched to branch 'JB'
```

```
E:\FULLSTACK\CloneEx2>git branch
  HPY
* JB
  master
```

4. Make a couple of commits in the branch – perhaps adding a new file and/or editing existing ones.

```
E:\FULLSTACK\CloneEx2>git status
On branch JB
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:   b1.txt

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

E:\FULLSTACK\CloneEx2>git commit -m "commit1"
On branch JB
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:   b1.txt

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

E:\FULLSTACK\CloneEx2>git add .

E:\FULLSTACK\CloneEx2>git commit -m "commit1"
[JB c67c0e8] commit1
 1 file changed, 1 insertion(+)
```

5. Use the log command to see the latest commits. The two you just made should be at the top of the list.

```
E:\FULLSTACK\CloneEx2>git log
commit c67c0e8ecd0c476890ac5c8daac513da30279bd5 (HEAD -> JB)
Author: Jb <jayabharath3600@gmail.com>
Date:   Fri Jul 12 17:38:25 2024 +0530

    commit1

commit 626bae11689a4039009f268778ad32da56a386cb (origin/master, origin/HEAD, master, HPY)
Author: Jb <jayabharath3600@gmail.com>
Date:   Fri Jul 12 16:13:11 2024 +0530

    ex2
```

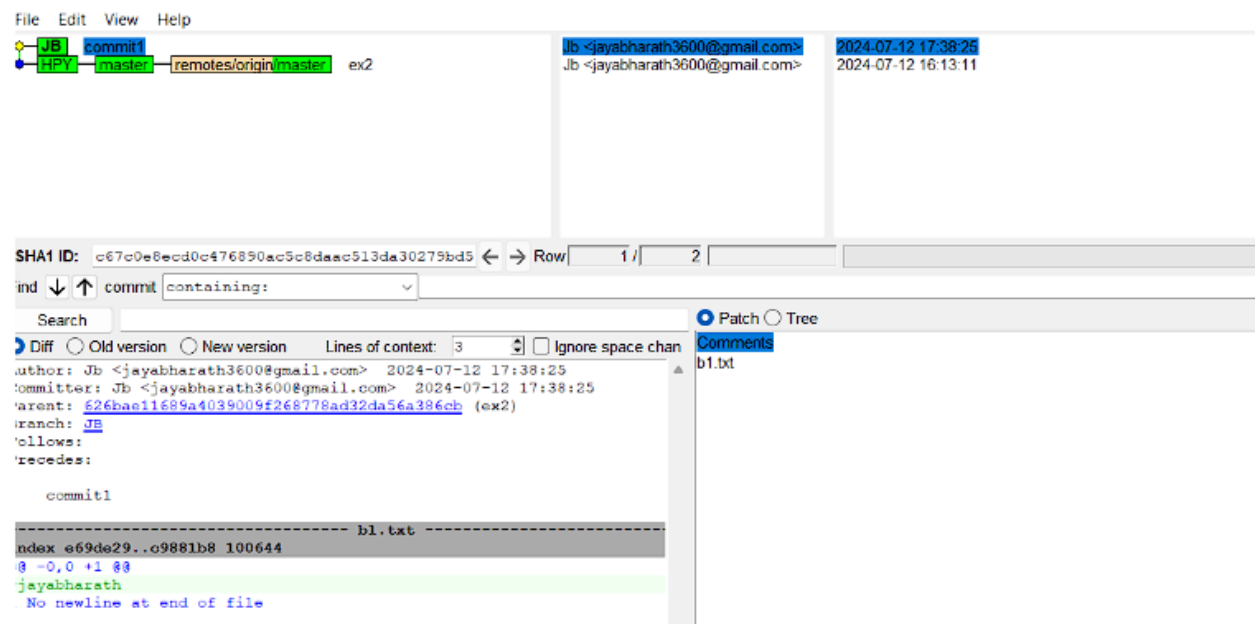
6. Use the checkout command to switch back to the master branch. Run log again. Notice your commits don't show up now. Check the files also – they should have their original contents.

```
E:\FULLSTACK\CloneEx2>git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

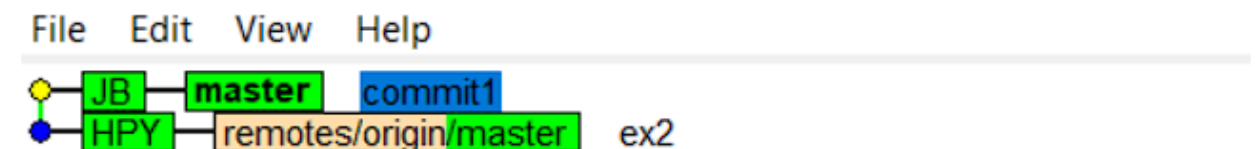
E:\FULLSTACK\CloneEx2>git log
commit 626bae11689a4039009f268778ad32da56a386cb (HEAD -> master, origin/master, origin/HEAD, HPY)
Author: Jb <jayabharath3600@gmail.com>
Date:   Fri Jul 12 16:13:11 2024 +0530

    ex2
```

7. Use the checkout command to switch back to your branch. Use gitk to take a look at the commit graph; notice it's linear.



8. Now checkout the master branch again. Use the merge command to merge your branch in to it. Look for information about it having been a fast-forward merge. Look at git log, and see that there is no merge commit. Take a look in gitk and see how the DAG is linear.



B

9. Switch back to your branch. Make a couple more commits.

```

E:\FULLSTACK\CloneEx2>git checkout JB
Switched to branch 'JB'

E:\FULLSTACK\CloneEx2>git add .

E:\FULLSTACK\CloneEx2>git commit -m "commit2"
[JB 465ca38] commit2
1 file changed, 1 insertion(+)

E:\FULLSTACK\CloneEx2>git add .

E:\FULLSTACK\CloneEx2>git commit -m "commit3"
[JB 52ee5cc] commit3
1 file changed, 3 insertions(+), 1 deletion(-)

```

10. Switch back to master. Make a commit there, which should edit a different file from the ones you touched in your branch – to be sure there is no conflict.

```

E:\FULLSTACK\CloneEx2>git commit -m "commit4"
[master ff5c10b] commit4
1 file changed, 1 insertion(+)
create mode 100644 b3.txt

E:\FULLSTACK\CloneEx2>git merge JB
hint: Waiting for your editor to close the file... C:\Users\Admin\AppData\Local\Programs\Microsoft VS Code\bin\code: lin
e 61: C:/Users/Admin/AppData/Local/Programs/Microsoft VS Code/Code.exe: Permission denied
error: there was a problem with the editor '"C:\Users\Admin\AppData\Local\Programs\Microsoft VS Code\bin\code" --wait'
Not committing merge; use 'git commit' to complete the merge.

E:\FULLSTACK\CloneEx2>git branch
  HPV
  JB
* master

E:\FULLSTACK\CloneEx2>git commit -m "commit5"
[master 1574001] commit5

E:\FULLSTACK\CloneEx2>git merge JB
Already up to date.

```

11. Now merge your branch again. (Aside: you don't need to do anything to inform Git that you only want to merge things added since your previous merge. Due to the way Git works, that kind of issue simply does not come up, unlike in early versions of Subversion.)

```

E:\FULLSTACK\CloneEx2>git log
commit 157400108d76dca916716eb7485715884e00feb2 (HEAD -> master)
Merge: ff5c10b 52ee5cc
Author: Jb <jayabharath3600@gmail.com>
Date:   Fri Jul 12 18:09:49 2024 +0530

    commit5

commit ff5c10b81ff5c4ee201a7243f117d058fdd61ee3
Author: Jb <jayabharath3600@gmail.com>
Date:   Fri Jul 12 18:06:55 2024 +0530

    commit4

commit 52ee5ccb881ca24a3323621c5e45d4576897d1ed (JB)
Author: Jb <jayabharath3600@gmail.com>
Date:   Fri Jul 12 18:05:06 2024 +0530

    commit3

commit 465ca3855688234dbd090a6a4057ca028051fa05
Author: Jb <jayabharath3600@gmail.com>
Date:   Fri Jul 12 18:04:24 2024 +0530

    commit2

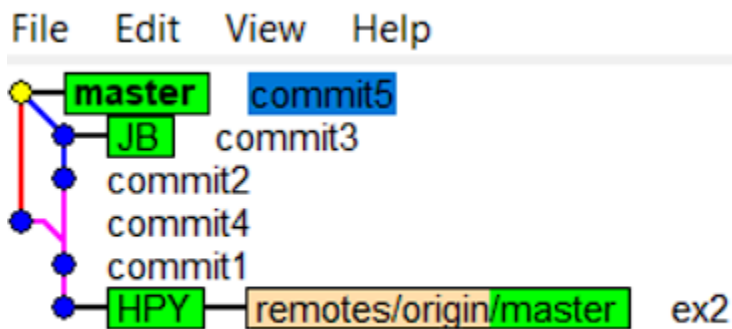
commit c67c0e8ecd0c476890ac5c8daac513da30279bd5
Author: Jb <jayabharath3600@gmail.com>
Date:   Fri Jul 12 17:38:25 2024 +0530

    commit1

commit 626bae11689a4039009f268778ad32da56a386cb (origin/master, origin/HEAD, HPY)
Author: Jb <jayabharath3600@gmail.com>
Date:   Fri Jul 12 16:13:11 2024 +0530

```

12. Look at git log. Notice that there is a merge commit. Also look in gitk.4



## EXERCISE 4

