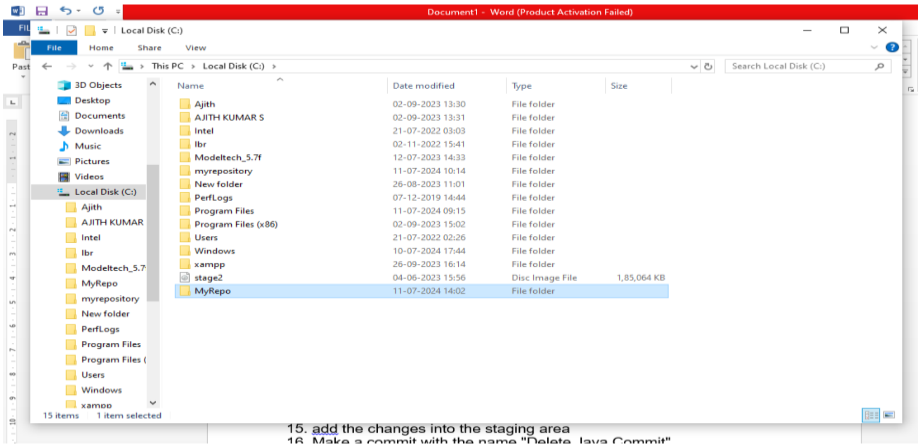
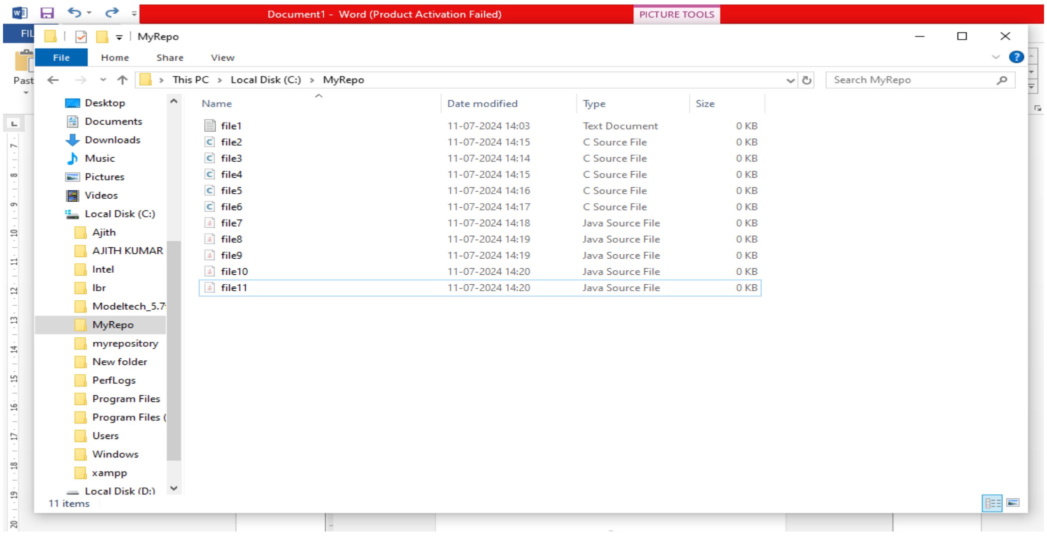
A. KEERTHANA 73152113053 cecskeerthanaacse25@gmail.com

1. Create a local git repo with the name "MyRepo".
2. create one .txt file, five .c files and five .java files in the working area.
3. Verify the git status
4. Add only the .txt into the staging area
5. verify the git status
6. Make a commit with the name "TXT Commit"
7. Add all .c files into the staging area
8. verify the git status
9. Make a commit with the name "C Commit"
10 .Add all .java files into the staging area
11. verify the git status
12. Make a commit with the name "Java Commit"
13. Delete all .java files from working area
14. Now see the status
15. add the changes into the staging area
16. Make a commit with the name "Delete Java Commit"
17. display all commit details

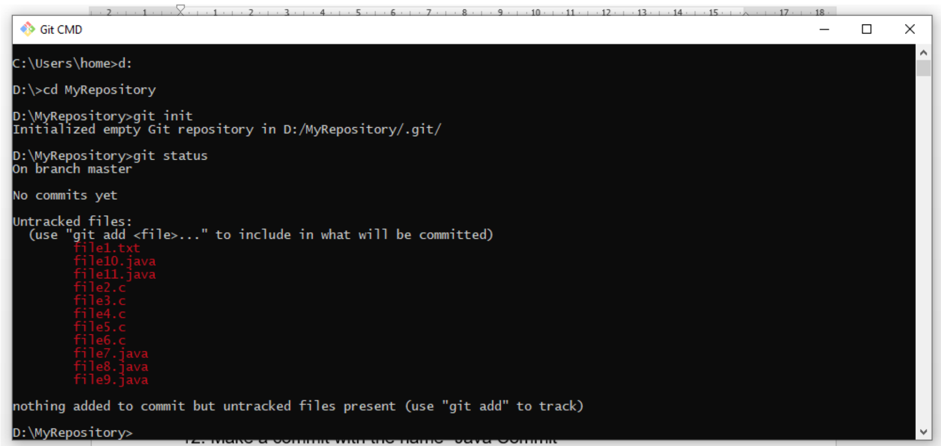


Name	Date modified	Type	Size
Ajith	02-09-2023 13:30	File folder	
AJITH KUMAR S	02-09-2023 13:31	File folder	
Intel	21-07-2022 03:03	File folder	
lbr	02-11-2022 15:41	File folder	
Modeltech_5.7f	12-07-2023 14:33	File folder	
myrepository	11-07-2024 10:14	File folder	
New folder	26-08-2023 11:01	File folder	
PerfLogs	07-12-2019 14:44	File folder	
Program Files	11-07-2024 09:15	File folder	
Program Files (x86)	02-09-2023 15:02	File folder	
Users	21-07-2022 02:26	File folder	
Windows	10-07-2024 17:44	File folder	
xampp	26-09-2023 16:14	File folder	
stage2	04-06-2023 15:56	Disc Image File	1,85,064 KB
MyRepo	11-07-2024 14:02	File folder	

15. add the changes into the staging area
16. Make a commit with the name "Delete Java Commit"

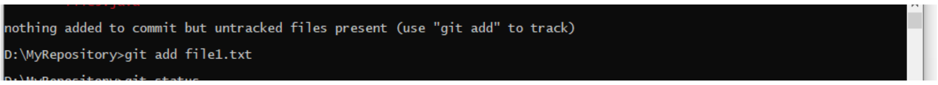


Name	Date modified	Type	Size
file1	11-07-2024 14:03	Text Document	0 KB
file2	11-07-2024 14:15	C Source File	0 KB
file3	11-07-2024 14:14	C Source File	0 KB
file4	11-07-2024 14:15	C Source File	0 KB
file5	11-07-2024 14:16	C Source File	0 KB
file6	11-07-2024 14:17	C Source File	0 KB
file7	11-07-2024 14:18	Java Source File	0 KB
file8	11-07-2024 14:19	Java Source File	0 KB
file9	11-07-2024 14:19	Java Source File	0 KB
file10	11-07-2024 14:20	Java Source File	0 KB
file11	11-07-2024 14:20	Java Source File	0 KB

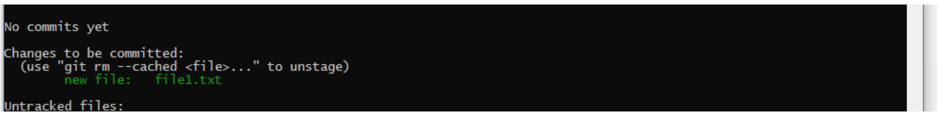


```
C:\Users\home>d:

D:\>cd MyRepository

D:\MyRepository>git init
Initialized empty Git repository in D:/MyRepository/.git/

D:\MyRepository>git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        file1.txt
        file10.java
        file11.java
        file2.c
        file3.c
        file4.c
        file5.c
        file6.c
        file7.java
        file8.java
        file9.java

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

D:\MyRepository>
```

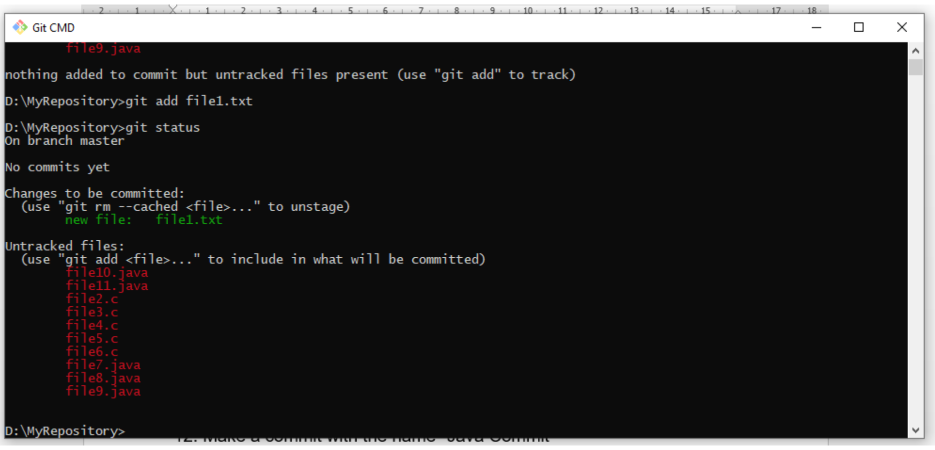
13. Delete all .java files from working area



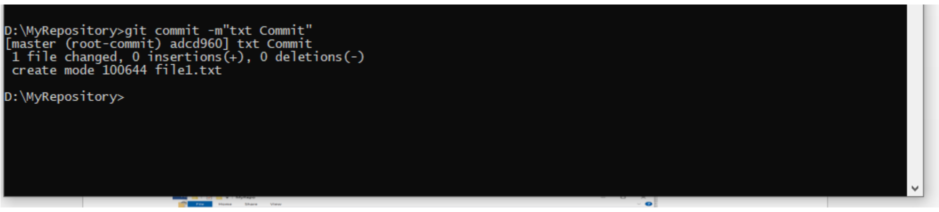
```
nothing added to commit but untracked files present (use "git add" to track)

D:\MyRepository>git add file1.txt

D:\MyRepository>git status
```



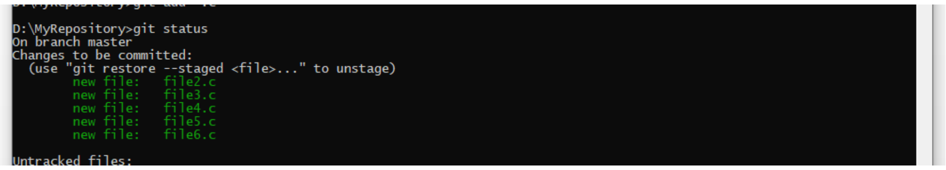
```
No commits yet

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

Untracked files:
```



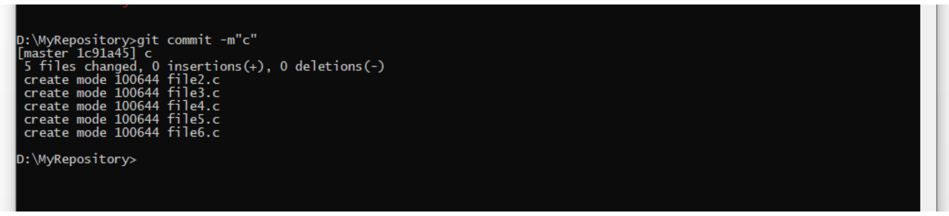
```
        file9.java

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

D:\MyRepository>git add file1.txt

D:\MyRepository>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)
        file10.java
        file11.java
        file2.c
        file3.c
        file4.c
        file5.c
        file6.c
        file7.java
        file8.java
        file9.java


D:\MyRepository>
```



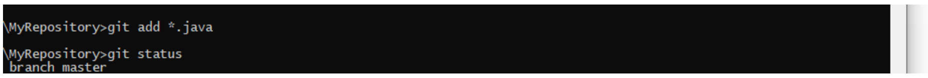
```
D:\MyRepository>git commit -m"txt Commit"
[master (root-commit) adcd960] txt Commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 file1.txt

D:\MyRepository>
```



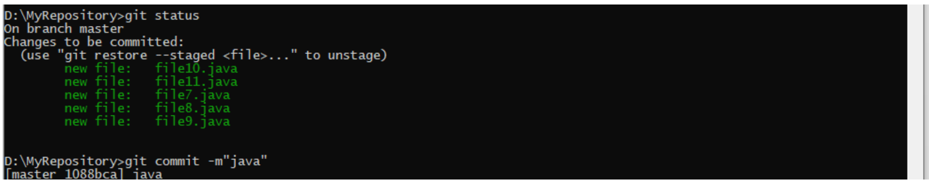
```
D:\MyRepository>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   file2.c
        new file:   file3.c
        new file:   file4.c
        new file:   file5.c
        new file:   file6.c

Untracked files:
```



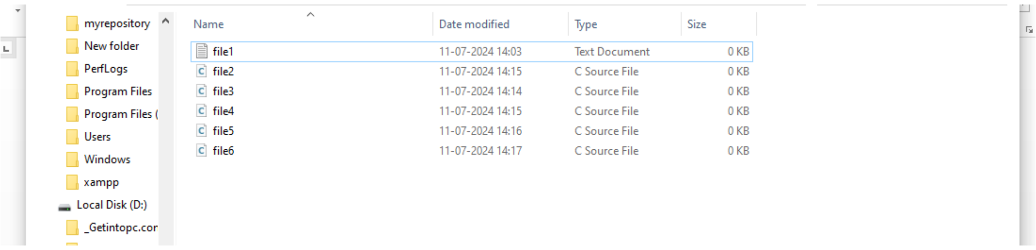
```
D:\MyRepository>git commit -m"c"
[master 1c91a45] c
 5 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 file2.c
 create mode 100644 file3.c
 create mode 100644 file4.c
 create mode 100644 file5.c
 create mode 100644 file6.c

D:\MyRepository>
```



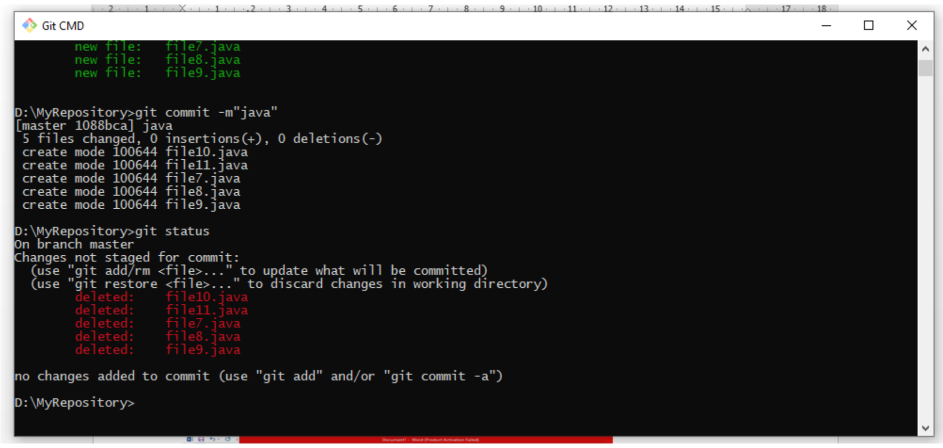
```
\MyRepository>git add *.java

\MyRepository>git status
 branch master
```



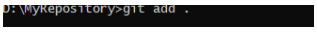
```
D:\MyRepository>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   file10.java
        new file:   file11.java
        new file:   file7.java
        new file:   file8.java
        new file:   file9.java


D:\MyRepository>git commit -m"java"
[master 1088bca] java
```



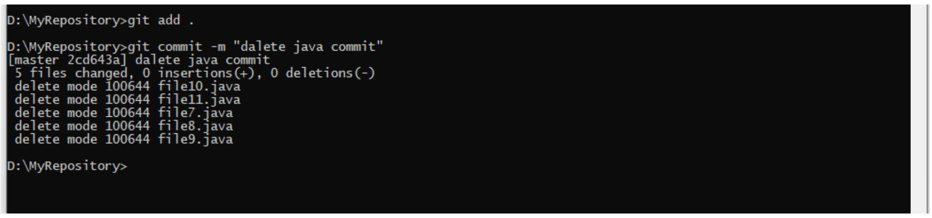
Name	Date modified	Type	Size
file1	11-07-2024 14:03	Text Document	0 KB
file2	11-07-2024 14:15	C Source File	0 KB
file3	11-07-2024 14:14	C Source File	0 KB
file4	11-07-2024 14:15	C Source File	0 KB
file5	11-07-2024 14:16	C Source File	0 KB
file6	11-07-2024 14:17	C Source File	0 KB



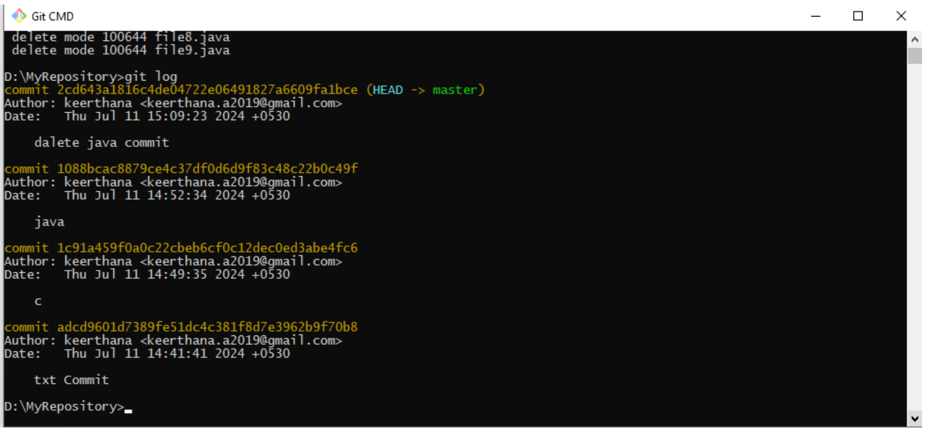
```
        new file:   file7.java
        new file:   file8.java
        new file:   file9.java

D:\MyRepository>git commit -m"java"
[master 1088bca] java
 5 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 file10.java
 create mode 100644 file11.java
 create mode 100644 file7.java
 create mode 100644 file8.java
 create mode 100644 file9.java

D:\MyRepository>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:    file10.java
        deleted:    file11.java
        deleted:    file7.java
        deleted:    file8.java
        deleted:    file9.java

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

D:\MyRepository>
```



```
D:\MyRepository>git add .
```



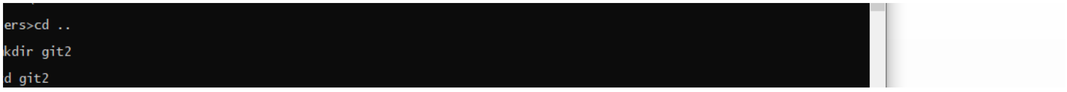
```
D:\MyRepository>git add .

D:\MyRepository>git commit -m "dalete java commit"
[master 2cd643a] dalete java commit
 5 files changed, 0 insertions(+), 0 deletions(-)
 delete mode 100644 file10.java
 delete mode 100644 file11.java
 delete mode 100644 file7.java
 delete mode 100644 file8.java
 delete mode 100644 file9.java

D:\MyRepository>
```



```
 delete mode 100644 file8.java
 delete mode 100644 file9.java

D:\MyRepository>git log
commit 2cd643a1816c4de04722e06491827a6609fa1bce (HEAD -> master)
Author: keerthana <keerthana.a2019@gmail.com>
Date:   Thu Jul 11 15:09:23 2024 +0530

    dalete java commit

commit 1088bcac8879ce4c37df0d6d9f83c48c22b0c49f
Author: keerthana <keerthana.a2019@gmail.com>
Date:   Thu Jul 11 14:52:34 2024 +0530

    java

commit 1c91a459f0a0c22cbeb6cf0c12dec0ed3abe4fc6
Author: keerthana <keerthana.a2019@gmail.com>
Date:   Thu Jul 11 14:49:35 2024 +0530

    c

commit adcd9601d7389fe51dc4c381f8d7e3962b9f70b8
Author: keerthana <keerthana.a2019@gmail.com>
Date:   Thu Jul 11 14:41:41 2024 +0530

    txt Commit

D:\MyRepository>
```

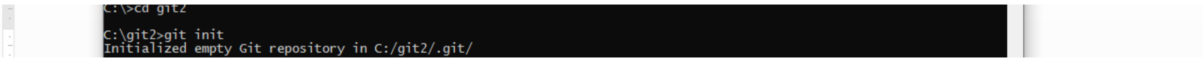
A.KEERTHANA 73152113053 cecskeerthanaa25@gmail.com

1. Create a new directory and change into it.



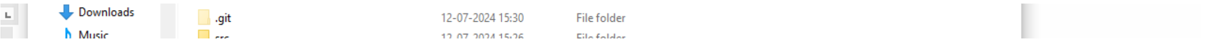
```
ers>cd ..
kdir git2
d git2
```

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



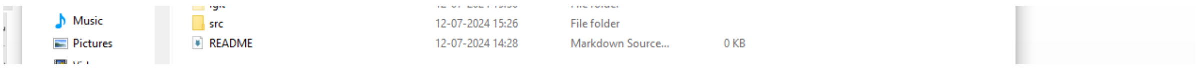
```
C:\>cd git2

C:\git2>git init
Initialized empty Git repository in C:/git2/.git/
```

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



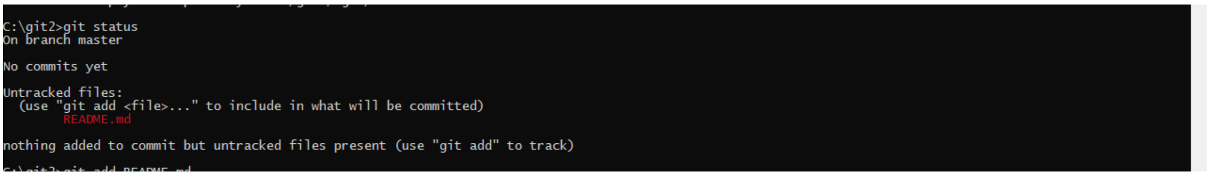
```
Downloads    .git    12-07-2024 15:30    File folder
Music
```

4. Create a README file.



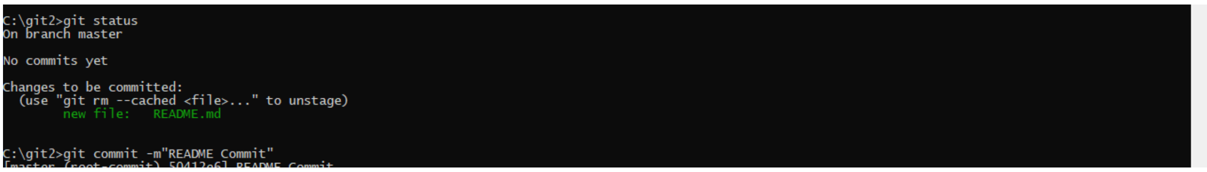
```
Music        src       12-07-2024 15:26    File folder
Pictures     README    12-07-2024 14:28    Markdown Source...    0 KB
```

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



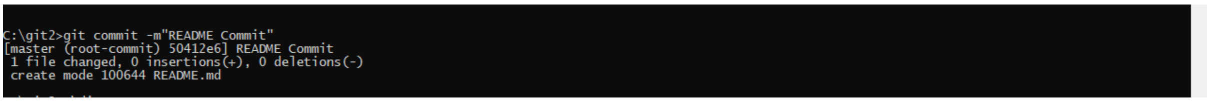
```
C:\git2>git status
On branch master

No commits yet

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

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

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



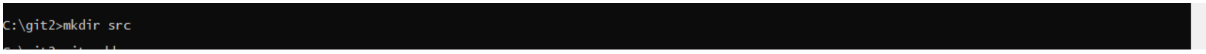
```
C:\git2>git status
On branch master

No commits yet

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


C:\git2>git commit -m"README Commit"
```

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



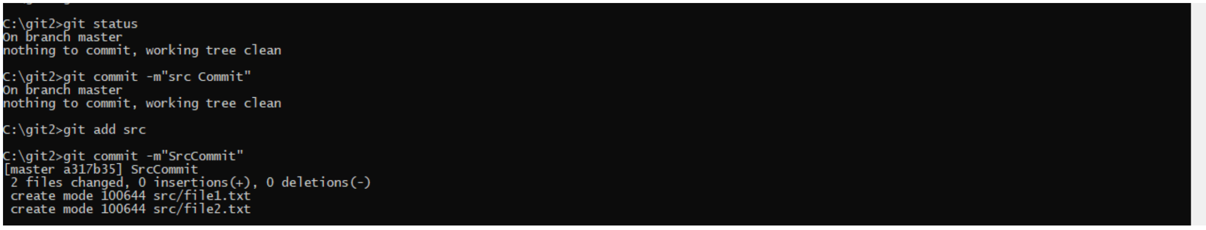
```
C:\git2>git commit -m"README Commit"
[master (root-commit) 50412e6] README Commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 README.md
```

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



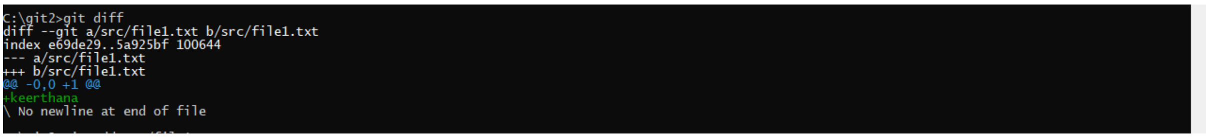
```
C:\git2>mkdir src
```

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.



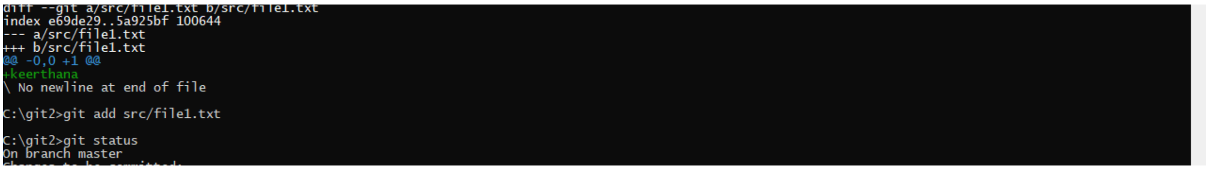
```
C:\git2>git status
On branch master
nothing to commit, working tree clean

C:\git2>git commit -m"src Commit"
On branch master
nothing to commit, working tree clean

C:\git2>git add src

C:\git2>git commit -m"SrcCommit"
[master a317b35] SrcCommit
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 src/file1.txt
 create mode 100644 src/file2.txt
```

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



```
C:\git2>git diff
diff --git a/src/file1.txt b/src/file1.txt
index e69de29..5a925bf 100644
--- a/src/file1.txt
+++ b/src/file1.txt
@@ -0,0 +1 @@
+keerthana
\ No newline at end of file
```

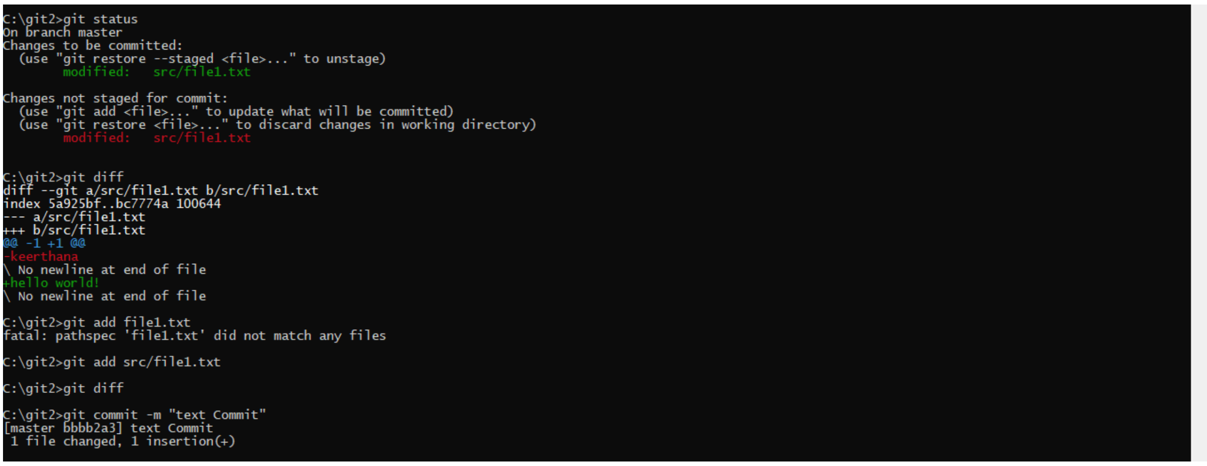
11. Next, add the changed file, and notice how it moves to the staging area in the status output. Also observe that the diff command you did before using add now gives no output. Why not? What do you have to do to see a diff of the things in the staging area? (Hint: review the slides if you can't remember.)



```
diff --git a/src/file1.txt b/src/file1.txt
index e69de29..5a925bf 100644
--- a/src/file1.txt
+++ b/src/file1.txt
@@ -0,0 +1 @@
+keerthana
\ No newline at end of file

C:\git2>git add src/file1.txt

C:\git2>git status
On branch master
```

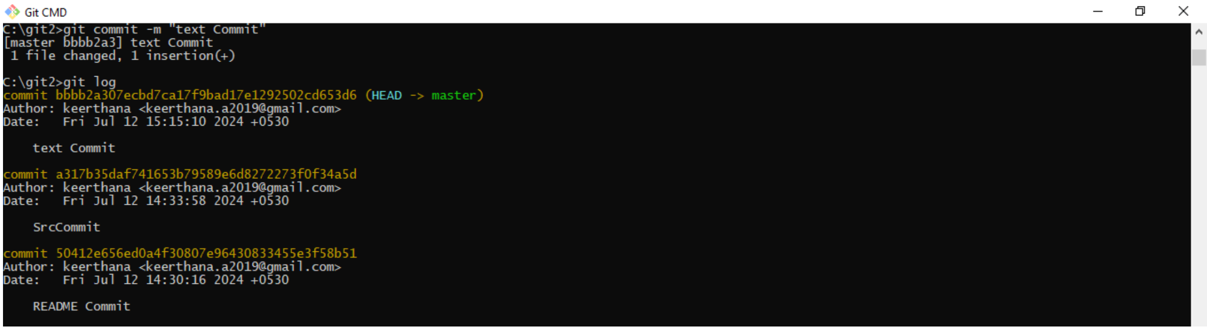
12. Now – without committing – make another change to the same file you changed in step 10. Look at the status output, and the diff output. Notice how you can have both staged and unstaged changes, even when you're talking about a single file. Observe the difference when you use the add command to stage the latest round of changes. Finally, commit them. You

should now have started to get a feel for the staging area.



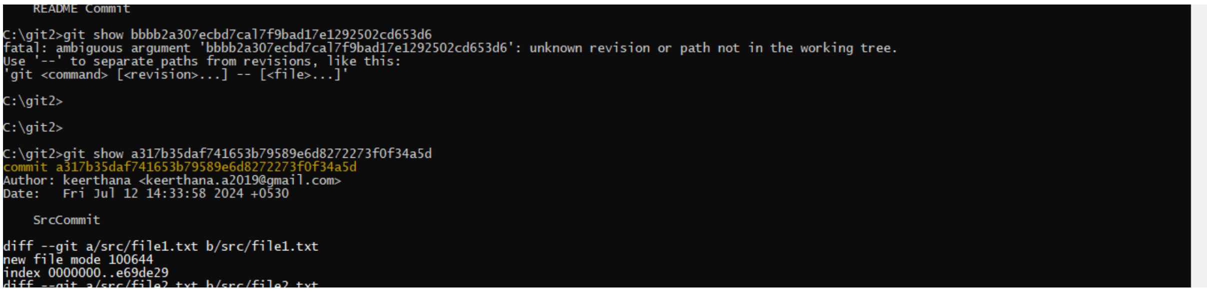
```
C:\git2>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   src/file1.txt

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:   src/file1.txt


C:\git2>git diff
diff --git a/src/file1.txt b/src/file1.txt
index 5a925bf..bc7774a 100644
--- a/src/file1.txt
+++ b/src/file1.txt
@@ -1 +1 @@
-keerthana
\ No newline at end of file
+hello world!
\ No newline at end of file

C:\git2>git add file1.txt
fatal: pathspec 'file1.txt' did not match any files

C:\git2>git add src/file1.txt

C:\git2>git diff

C:\git2>git commit -m "text Commit"
[master bbbb2a3] text Commit
 1 file changed, 1 insertion(+)
```

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



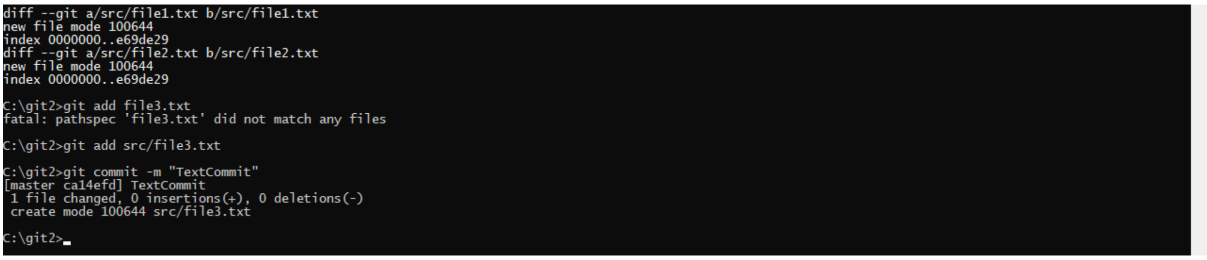
```
C:\git2>git commit -m "text Commit"
[master bbbb2a3] text Commit
 1 file changed, 1 insertion(+)

C:\git2>git log
commit bbbb2a307ecbd7ca17f9bad17e1292502cd653d6 (HEAD -> master)
Author: keerthana <keerthana.a2019@gmail.com>
Date:   Fri Jul 12 15:15:10 2024 +0530

    text Commit

commit a317b35daf741653b79589e6d8272273f0f34a5d
Author: keerthana <keerthana.a2019@gmail.com>
Date:   Fri Jul 12 14:33:58 2024 +0530

    SrcCommit

commit 50412e656ed0a4f30807e96430833455e3f58b51
Author: keerthana <keerthana.a2019@gmail.com>
Date:   Fri Jul 12 14:30:16 2024 +0530

    README Commit
```

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?



```
    README Commit

C:\git2>git show bbbb2a307ecbd7cal7f9bad17e1292502cd653d6
fatal: ambiguous argument 'bbbb2a307ecbd7cal7f9bad17e1292502cd653d6': unknown revision or path not in the working tree.
Use '--' to separate paths from revisions, like this:
'git <command> [<revision>...] -- [<file>...]'

C:\git2>

C:\git2>

C:\git2>git show a317b35daf741653b79589e6d8272273f0f34a5d
commit a317b35daf741653b79589e6d8272273f0f34a5d
Author: keerthana <keerthana.a2019@gmail.com>
Date:   Fri Jul 12 14:33:58 2024 +0530

    SrcCommit

diff --git a/src/file1.txt b/src/file1.txt
new file mode 100644
index 0000000..e69de29
diff --git a/src/file2.txt b/src/file2.txt
```

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

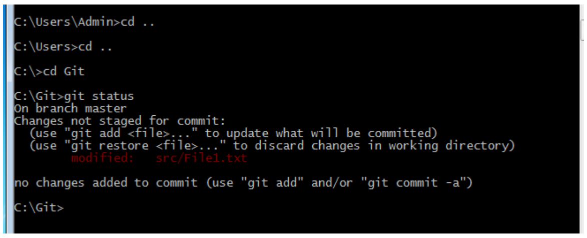


```
diff --git a/src/file1.txt b/src/file1.txt
new file mode 100644
index 0000000..e69de29
diff --git a/src/file2.txt b/src/file2.txt
new file mode 100644
index 0000000..e69de29

C:\git2>git add file3.txt
fatal: pathspec 'file3.txt' did not match any files

C:\git2>git add src/file3.txt

C:\git2>git commit -m "TextCommit"
[master ca14efd] TextCommit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 src/file3.txt

C:\git2>
```

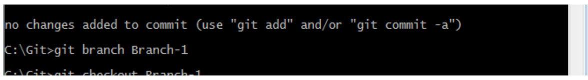
1. Run the status command. Notice how it tells you what branch you are in.
2. Use the branch command to create a new branch.
3. Use the checkout command to switch to it.
4. Make a couple of commits in the branch – perhaps adding a new file and/or editing existing ones.
5. Use the log command to see the latest commits. The two you just made should be at the top of the list.
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.
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.
9. Switch back to your branch. Make a couple more commits.
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.
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.)

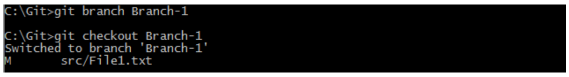
12. Look at git log. Notice that there is a merge commit. Also look in gitk. Notice the DAG
now shows how things forked, and then were joined up again by a merge commit.



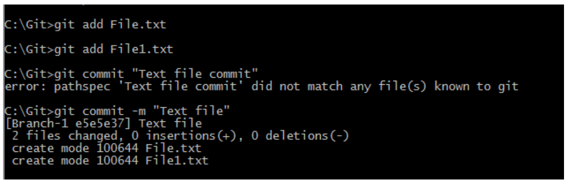
```
C:\Users\Admin>cd ..

C:\Users>cd ..

C:\>cd Git

C:\Git>git status
On branch master
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:   src/File1.txt

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

C:\Git>
```



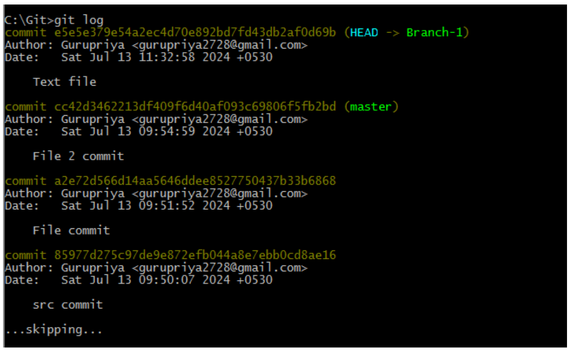
```
no changes added to commit (use "git add" and/or "git commit -a")

C:\Git>git branch Branch-1

C:\Git>git checkout Branch-1
```



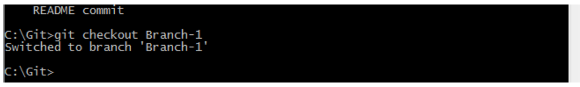
```
C:\Git>git branch Branch-1

C:\Git>git checkout Branch-1
Switched to branch 'Branch-1'
M       src/File1.txt
```



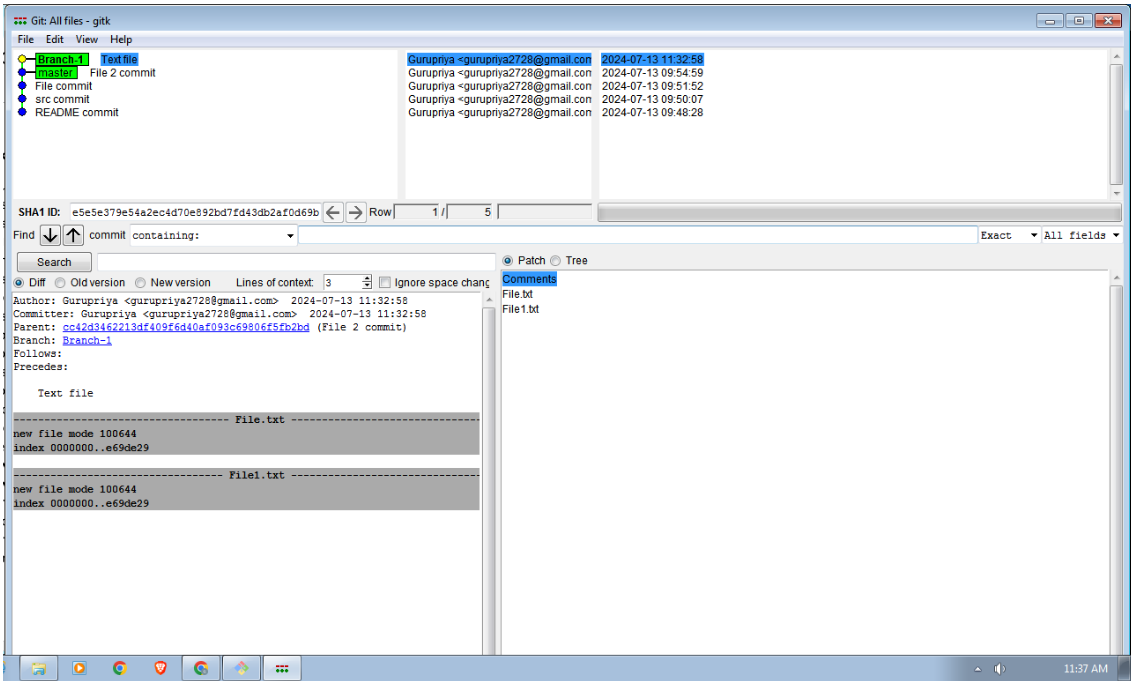
```
C:\Git>git add File.txt

C:\Git>git add File1.txt

C:\Git>git commit "Text file commit"
error: pathspec 'Text file commit' did not match any file(s) known to git

C:\Git>git commit -m "Text file"
[Branch-1 e5e5e37] Text file
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 File.txt
 create mode 100644 File1.txt
```



```
C:\Git>git log
commit e5e5e379e54a2ec4d70e892bd7fd43db2af0d69b (HEAD -> Branch-1)
Author: Gurupriya <gurupriya2728@gmail.com>
Date:   Sat Jul 13 11:32:58 2024 +0530

    Text file

commit cc42d3462213df409f6d40af093c69806f5fb2bd (master)
Author: Gurupriya <gurupriya2728@gmail.com>
Date:   Sat Jul 13 09:54:59 2024 +0530

    File 2 commit

commit a2e72d566d14aa5646ddee8527750437b33b6868
Author: Gurupriya <gurupriya2728@gmail.com>
Date:   Sat Jul 13 09:51:52 2024 +0530

    File commit

commit 85977d275c97de9e872efb044a8e7ebb0cd8ae16
Author: Gurupriya <gurupriya2728@gmail.com>
Date:   Sat Jul 13 09:50:07 2024 +0530

    src commit

...skipping...
```



```
    README commit

C:\Git>git checkout Branch-1
Switched to branch 'Branch-1'

C:\Git>
```



```
Branch-1  Text file            Gurupriya <gurupriya2728@gmail.com  2024-07-13 11:32:58
master  File 2 commit          Gurupriya <gurupriya2728@gmail.com  2024-07-13 09:54:59
File commit                    Gurupriya <gurupriya2728@gmail.com  2024-07-13 09:51:52
src commit                     Gurupriya <gurupriya2728@gmail.com  2024-07-13 09:50:07
README commit                  Gurupriya <gurupriya2728@gmail.com  2024-07-13 09:48:28
```

SHA1 ID: e5e5e379e54a2ec4d70e892bd7fd43db2af0d69b Row 1 / 5

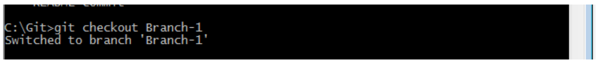
commit containing:

● Patch ○ Tree

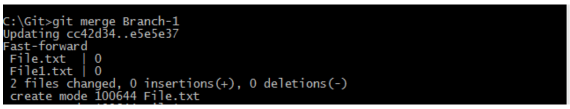
● Diff ○ Old version ○ New version Lines of context: 3 □ Ignore space chang

```
Author: Gurupriya <gurupriya2728@gmail.com>  2024-07-13 11:32:58
Committer: Gurupriya <gurupriya2728@gmail.com>  2024-07-13 11:32:58
Parent: cc42d3462213df409f6d40af093c69806f5fb2bd (File 2 commit)
Branch: Branch-1
Follows:
Precedes:

    Text file

---------------------------------- File.txt ----------------------------------
new file mode 100644
index 0000000..e69de29

--------------------------------- File1.txt ----------------------------------
new file mode 100644
index 0000000..e69de29
```

Comments
File.txt
File1.txt



```
C:\Git>git checkout Branch-1
Switched to branch 'Branch-1'
```

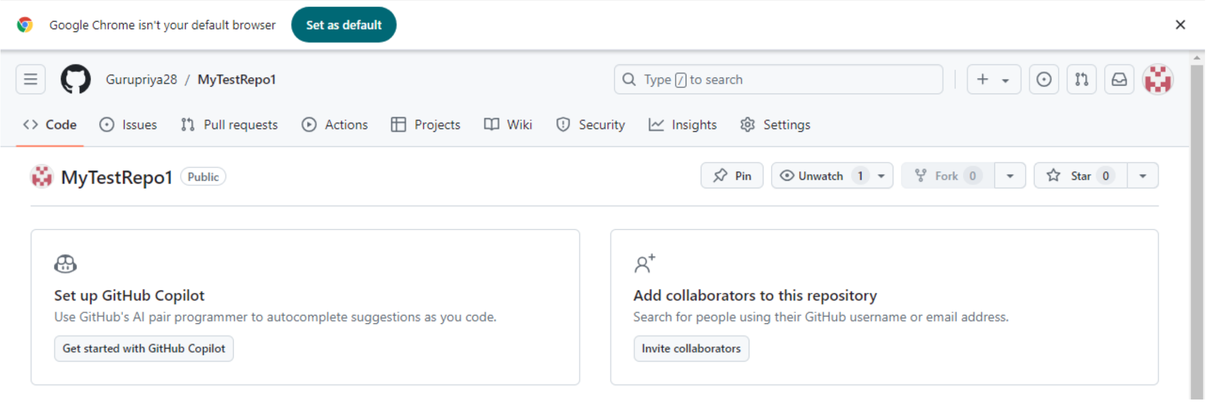


```
C:\Git>git merge Branch-1
Updating cc42d34..e5e5e37
Fast-forward
 File.txt  | 0
 File1.txt | 0
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 File.txt
```

Exercise 3

Main Task

1. First, one person in the group should create a public repository using their GitHub account.



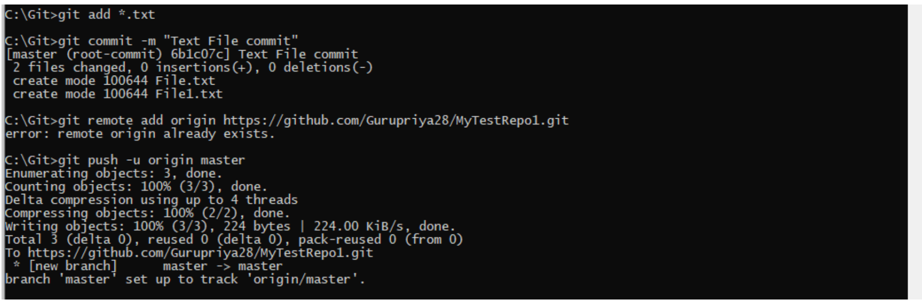
Set up GitHub Copilot

Use GitHub's AI pair programmer to autocomplete suggestions as you code.

Add collaborators to this repository

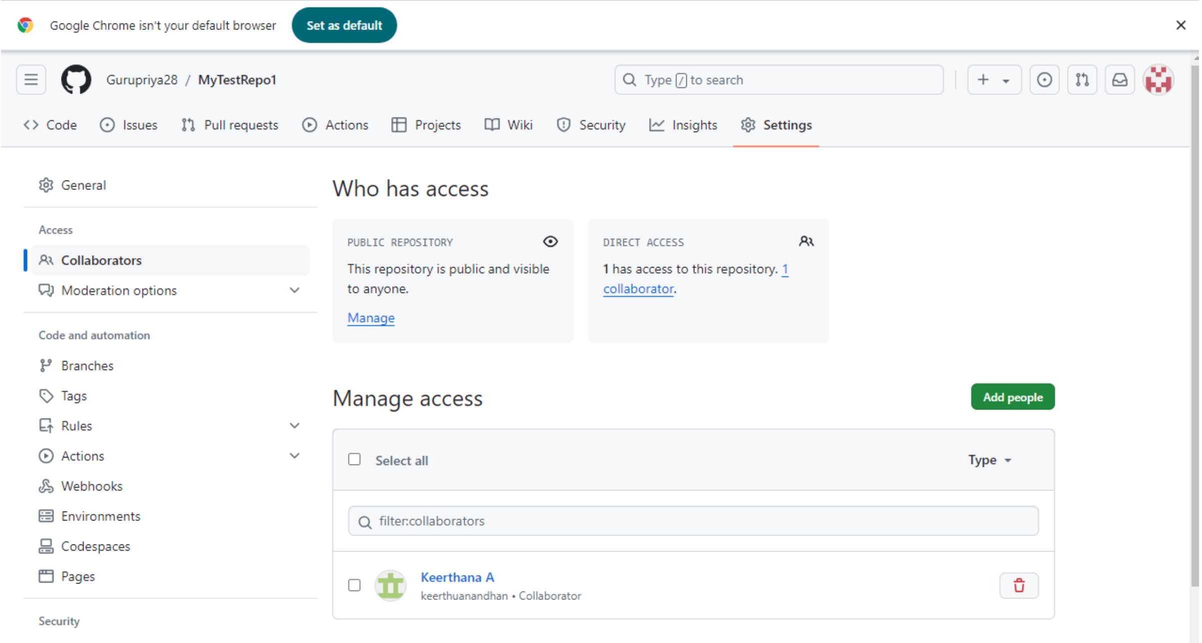
Search for people using their GitHub username or email address.

2. This same person should then follow the instructions from GitHub to add a remote, and then push their repository. Do not forget the –u flag, as suggested by GitHub!



```
C:\Git>git add *.txt

C:\Git>git commit -m "Text File commit"
[master (root-commit) 6b1c07c] Text File commit
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 File.txt
 create mode 100644 File1.txt

C:\Git>git remote add origin https://github.com/Gurupriya28/MyTestRepo1.git
error: remote origin already exists.

C:\Git>git push -u origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 224 bytes | 224.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Gurupriya28/MyTestRepo1.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
```

3. All of the other members of the group should then be added as collaborators, so they can commit to the repository also.



Who has access

PUBLIC REPOSITORY

This repository is public and visible to anyone.

DIRECT ACCESS

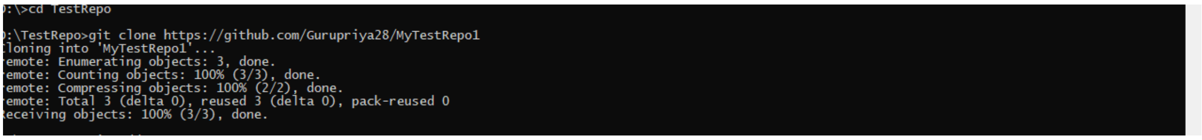
1 has access to this repository. 1 collaborator.

Manage access

□ Keerthana A

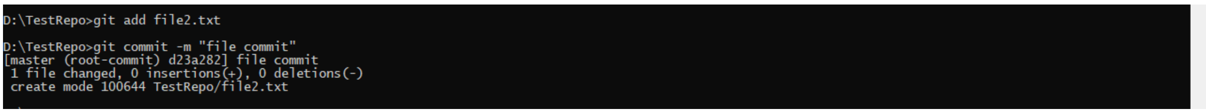
keerthuanandhan • Collaborator

4. Next, everyone else in the group should clone the repository from GitHub. Verify that the context of the repository is what is expected.

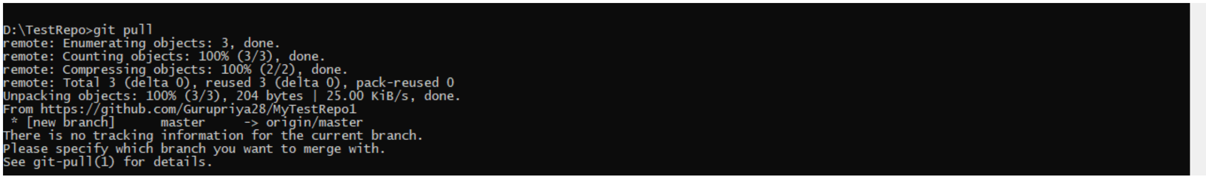


```
D:\>cd TestRepo

D:\TestRepo>git clone https://github.com/Gurupriya28/MyTestRepo1
Cloning into 'MyTestRepo1'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
```

5. One of the group members who just cloned should now make a local commit, then push it. Everyone should verify that when they pull, that commit is added to their local repository (use git log to check for it).

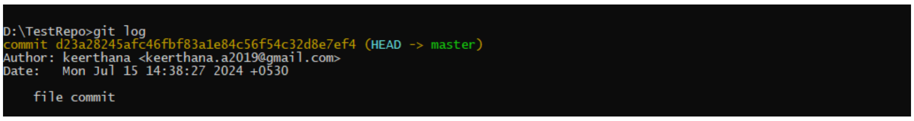


```
D:\TestRepo>git add file2.txt

D:\TestRepo>git commit -m "file commit"
[master (root-commit) d23a282] file commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 TestRepo/file2.txt
```



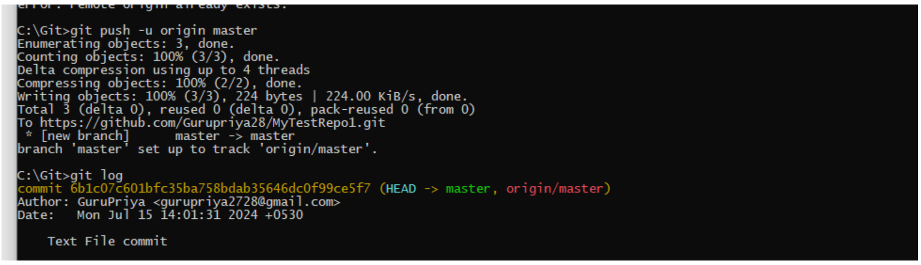
```
D:\TestRepo>git pull
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 204 bytes | 25.00 KiB/s, done.
From https://github.com/Gurupriya28/MyTestRepo1
 * [new branch]      master     -> origin/master
There is no tracking information for the current branch.
Please specify which branch you want to merge with.
See git-pull(1) for details.
```

6. Look at each other's git log output. Notice how the SHA-1 is the same for a given commit across every copy of the repository. Why is this important?



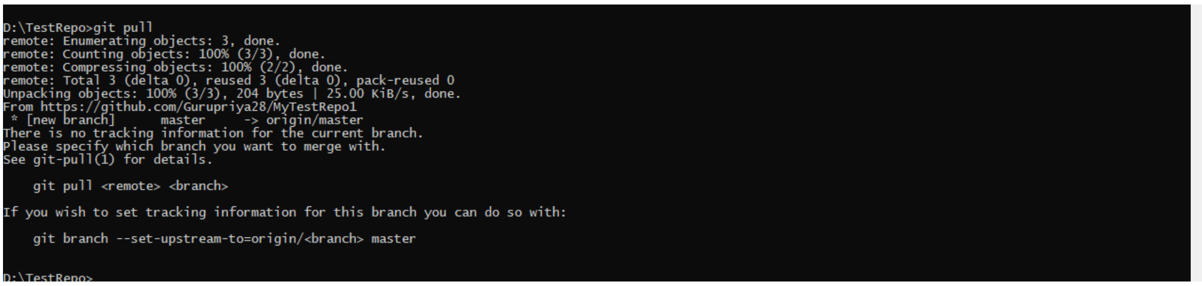
```
D:\TestRepo>git log
commit d23a28245afc46fbf83a1e84c56f54c32d8e7ef4 (HEAD -> master)
Author: keerthana <keerthana.a2019@gmail.com>
Date:   Mon Jul 15 14:38:27 2024 +0530

    file commit
```

7. Two members of the group should now make a commit locally, and race to push it. To keep things simple, be sure to edit different files. What happens to the runner-up?



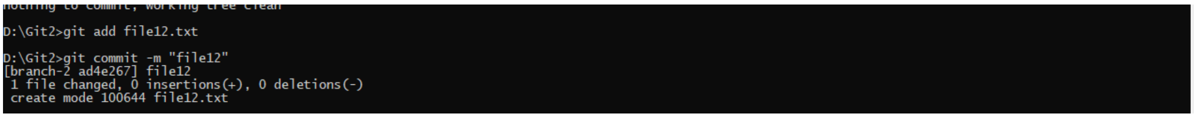
```
error: remote origin already exists.

C:\Git>git push -u origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 224 bytes | 224.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Gurupriya28/MyTestRepo1.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.

C:\Git>git log
commit 6b1c07c601bfc35ba758bdab35646dc0f99ce5f7 (HEAD -> master, origin/master)
Author: GuruPriya <gurupriya2728@gmail.com>
Date:   Mon Jul 15 14:01:31 2024 +0530

    Text File commit
```

8. The runner-up should now pull. As a group, look at the output of the command. Additionally, look at the git log, and notice that there is a merge commit. You may also wish to view the DAG in gitk.

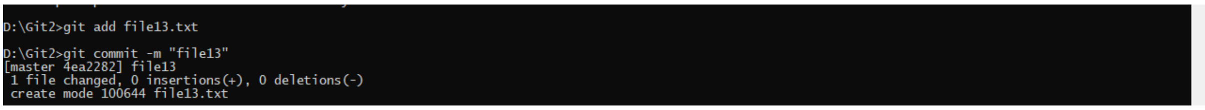


```
D:\TestRepo>git pull
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 204 bytes | 25.00 KiB/s, done.
From https://github.com/Gurupriya28/MyTestRepo1
 * [new branch]      master     -> origin/master
There is no tracking information for the current branch.
Please specify which branch you want to merge with.
See git-pull(1) for details.

    git pull <remote> <branch>

If you wish to set tracking information for this branch you can do so with:

    git branch --set-upstream-to=origin/<branch> master

D:\TestRepo>
```

9. Repeat the last two steps a couple of times, to practice.



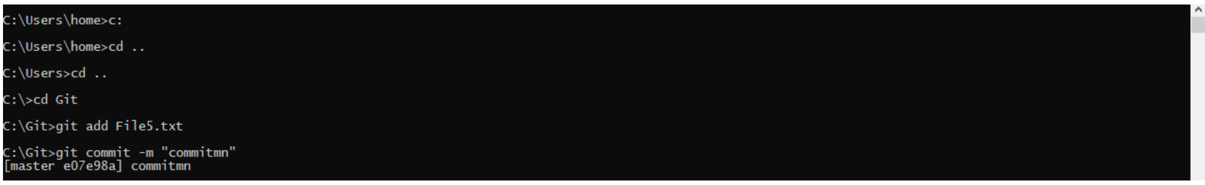
```
nothing to commit, working tree clean

D:\Git2>git add file12.txt

D:\Git2>git commit -m "file12"
[branch-2 ad4e267] file12
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 file12.txt
```



```
D:\Git2>git add file13.txt

D:\Git2>git commit -m "file13"
[master 4ea2282] file13
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 file13.txt
```

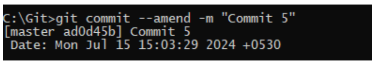
Exercise 4

1. Make a commit, and make a silly typo in the commit message.



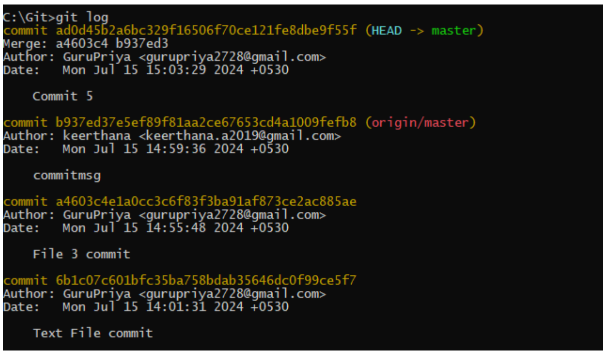
```
C:\Users\home>c:

C:\Users\home>cd ..

C:\Users>cd ..

C:\>cd Git

C:\Git>git add File5.txt

C:\Git>git commit -m "commitmn"
[master e07e98a] commitmn
```

2. Use the --amend flag to enable you to fix the commit message.



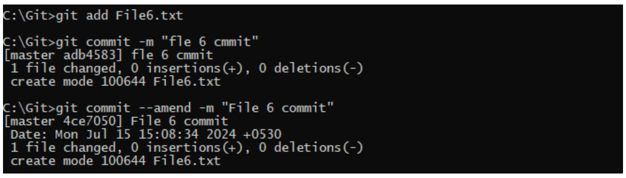
```
C:\Git>git commit --amend -m "Commit 5"
[master ad0d45b] Commit 5
 Date: Mon Jul 15 15:03:29 2024 +0530
```

3. Look at the log and notice how the mistake is magically gone.



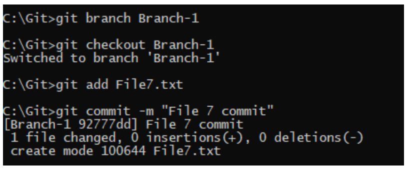
```
C:\Git>git log
commit ad0d45b2a6bc329f16506f70ce121fe8dbe9f55f (HEAD -> master)
Merge: a4603c4 b937ed3
Author: GuruPriya <gurupriya2728@gmail.com>
Date:   Mon Jul 15 15:03:29 2024 +0530

    Commit 5

commit b937ed37e5ef89f81aa2ce67653cd4a1009fefb8 (origin/master)
Author: keerthana <keerthana.a2019@gmail.com>
Date:   Mon Jul 15 14:59:36 2024 +0530

    commitmsg

commit a4603c4e1a0cc3c6f83f3ba91af873ce2ac885ae
Author: GuruPriya <gurupriya2728@gmail.com>
Date:   Mon Jul 15 14:55:48 2024 +0530

    File 3 commit

commit 6b1c07c601bfc35ba758bdab35646dc0f99ce5f7
Author: GuruPriya <gurupriya2728@gmail.com>
Date:   Mon Jul 15 14:01:31 2024 +0530

    Text File commit
```

4. Now make a commit where you make a typo in one of the files. Once again, use --amend to magic away your problems.



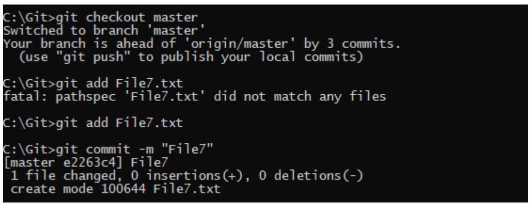
```
C:\Git>git add File6.txt

C:\Git>git commit -m "fle 6 cmmit"
[master adb4583] fle 6 cmmit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 File6.txt

C:\Git>git commit --amend -m "File 6 commit"
[master 4ce7050] File 6 commit
 Date: Mon Jul 15 15:08:34 2024 +0530
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 File6.txt
```

5. Create a branch. Make a commit.



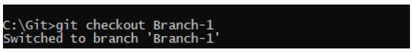
```
C:\Git>git branch Branch-1

C:\Git>git checkout Branch-1
Switched to branch 'Branch-1'

C:\Git>git add File7.txt

C:\Git>git commit -m "File 7 commit"
[Branch-1 92777dd] File 7 commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 File7.txt
```

6. Now switch back to your master branch. Make a (non-conflicting) commit there also.



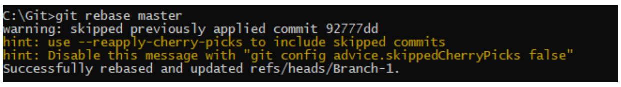
```
C:\Git>git checkout master
Switched to branch 'master'
Your branch is ahead of 'origin/master' by 3 commits.
  (use "git push" to publish your local commits)

C:\Git>git add File7.txt
fatal: pathspec 'File7.txt' did not match any files

C:\Git>git add File7.txt

C:\Git>git commit -m "File7"
[master e2263c4] File7
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 File7.txt
```

7. Now switch back to your branch.



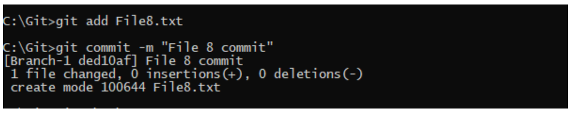
```
C:\Git>git checkout Branch-1
Switched to branch 'Branch-1'
```

8. Use the rebase command in your branch. Look at the DAG in gitk, and note that you have the commit from the master branch, but no merge commit.



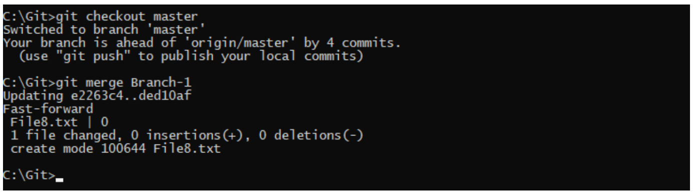
```
C:\Git>git rebase master
warning: skipped previously applied commit 92777dd
hint: use --reapply-cherry-picks to include skipped commits
hint: Disable this message with "git config advice.skippedCherryPicks false"
Successfully rebased and updated refs/heads/Branch-1.
```

9. Make one more commit in your branch.



```
C:\Git>git add File8.txt

C:\Git>git commit -m "File 8 commit"
[Branch-1 ded10af] File 8 commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 File8.txt
```

10. Return to master. Merge your branch. Notice how, thanks to the rebase, this is a fastforward merge.



```
C:\Git>git checkout master
Switched to branch 'master'
Your branch is ahead of 'origin/master' by 4 commits.
  (use "git push" to publish your local commits)

C:\Git>git merge Branch-1
Updating e2263c4..ded10af
Fast-forward
 File8.txt | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 File8.txt

C:\Git>
```