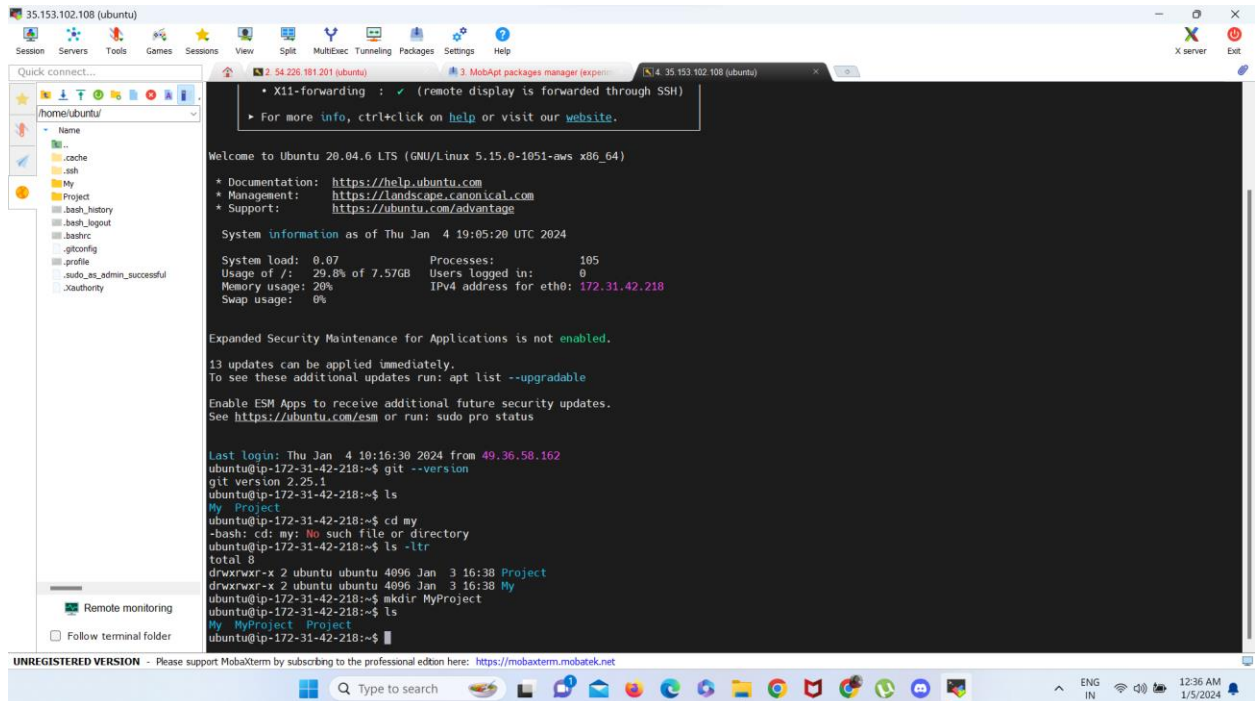
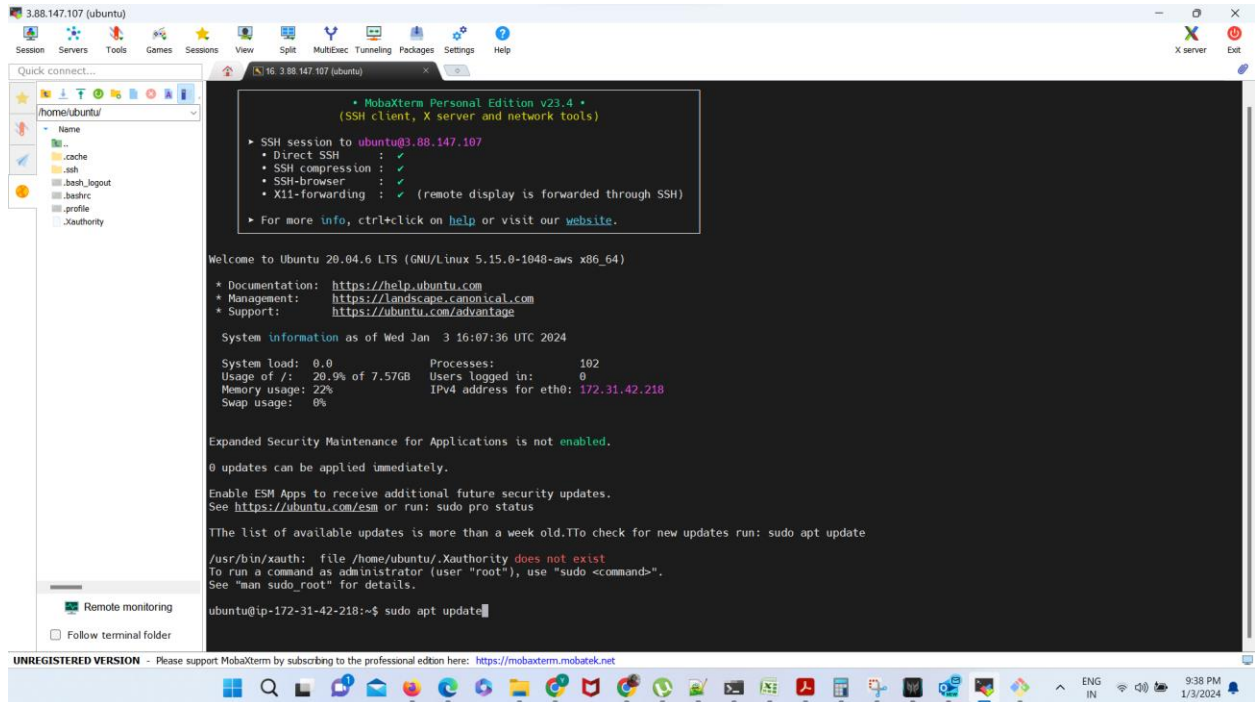
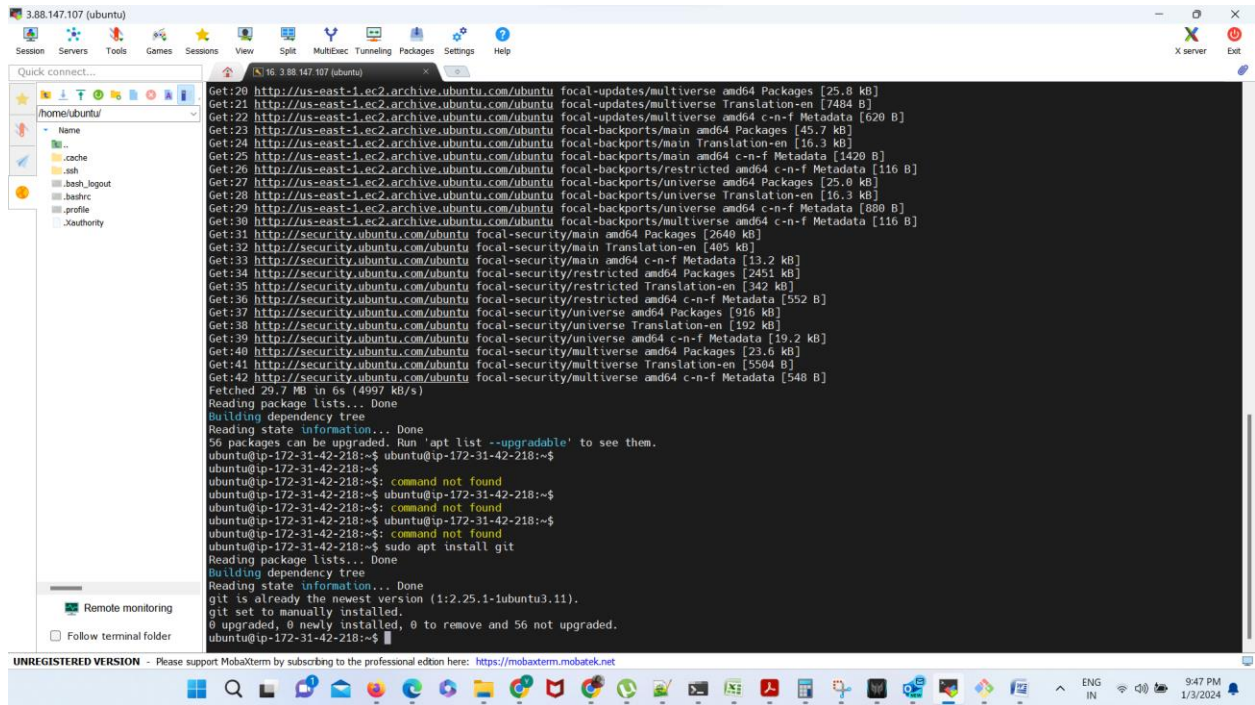
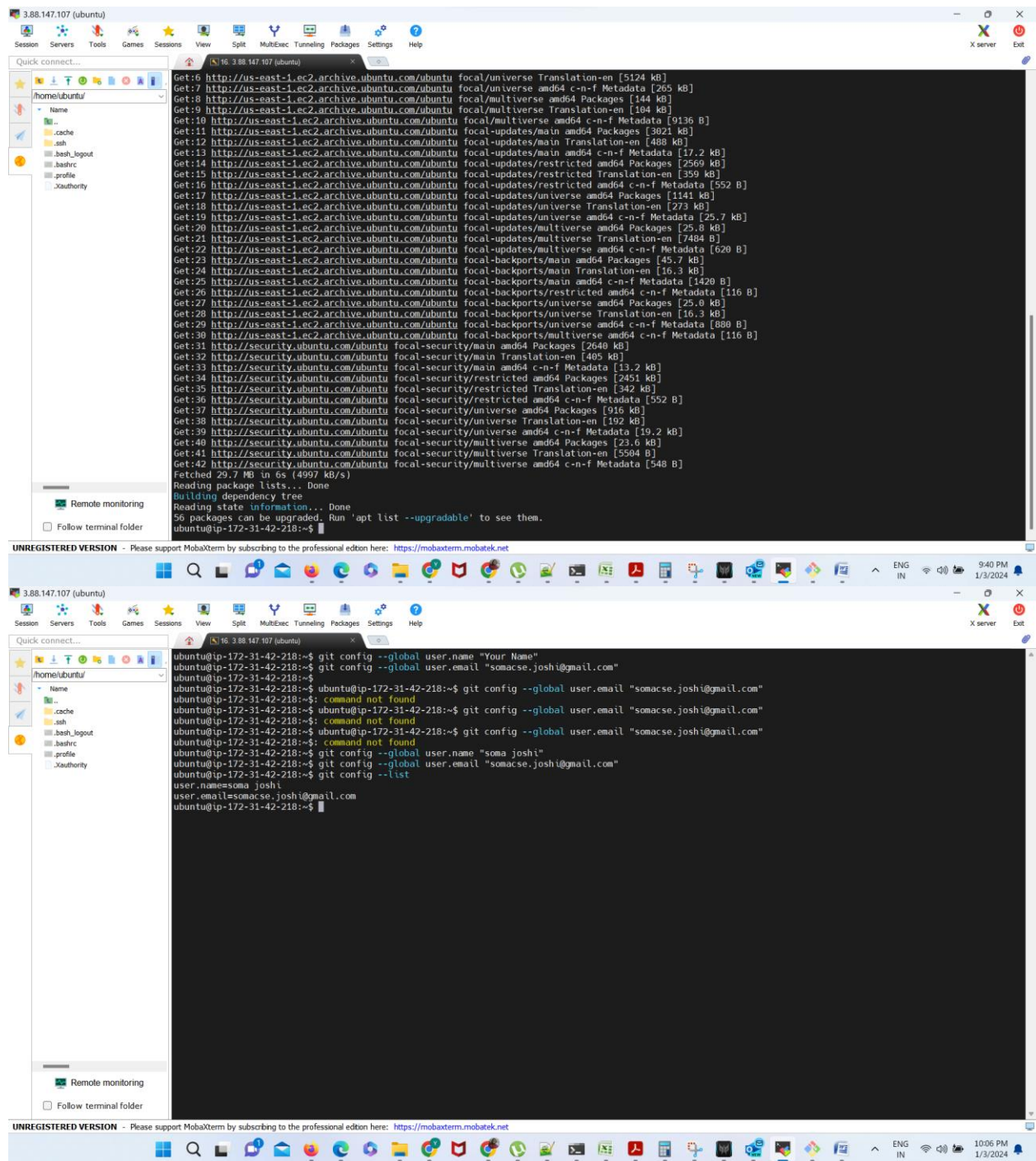


Assignment -1








```
35.153.102.108 (ubuntu)
Session Servers Tools Games Sessions View Split Multitab Tunneling Packages Settings Help
Quick connect...
/home/ubuntu/
Name
.ccache
.cssh
.cssh
My
Project
.bash_history
.bash_logout
.bashrc
.gitconfig
.gitignore
.profile
.sudo_as_admin_successful
.xauthority
Remote monitoring
Follow terminal folder
root@ip-172-31-42-218:~# cd / My
My/
My Project/
root@ip-172-31-42-218:~# exit
logout
ubuntu@ip-172-31-42-218:/$ ls
bin boot dev etc home lib lib32 lib64 libx32 lost+found media mnt opt proc root run sbin snap srv sys usr var
ubuntu@ip-172-31-42-218:/$ ls -l
total 64
lrwxrwxrwx 1 root root 7 Oct 25 21:51 bin -> usr/bin
drwxr-xr-x 4 root root 4096 Jan 4 19:02 boot
drwxr-xr-x 17 root root 3220 Jan 4 19:02 dev
drwxr-xr-x 95 root root 4096 Jan 4 19:02 etc
drwxr-xr-x 3 root root 4096 Jan 3 15:41 home
lrwxrwxrwx 1 root root 7 Oct 25 21:51 lib -> usr/lib
lrwxrwxrwx 1 root root 9 Oct 25 21:51 lib32 -> usr/lib32
lrwxrwxrwx 1 root root 9 Oct 25 21:51 lib64 -> usr/lib64
lrwxrwxrwx 1 root root 10 Oct 25 21:51 libx32 -> usr/libx32
drwx----- 2 root root 16384 Oct 25 21:54 lost+found
drwxr-xr-x 2 root root 4096 Oct 25 21:51 media
drwxr-xr-x 2 root root 4096 Oct 25 21:51 mnt
drwxr-xr-x 2 root root 4096 Oct 25 21:51 opt
dr-xr-xr-x 160 root root 0 Jan 4 19:02 proc
drwx----- 7 root root 4096 Jan 3 20:55 root
drwxr-xr-x 25 root root 860 Jan 4 19:05 run
lrwxrwxrwx 1 root root 8 Oct 25 21:51 sbin -> usr/sbin
drwxr-xr-x 8 root root 4096 Oct 25 21:56 snap
drwxr-xr-x 2 root root 4096 Oct 25 21:51 srv
dr-xr-xr-x 13 root root 0 Jan 4 19:02 sys
drwxrwxrwt 12 root root 4096 Jan 4 19:05 tmp
drwxr-xr-x 14 root root 4096 Oct 25 21:52 usr
drwxr-xr-x 13 root root 4096 Oct 25 21:53 var
ubuntu@ip-172-31-42-218:/$ cd ~
ubuntu@ip-172-31-42-218:~$ ls
My MyProject Project
ubuntu@ip-172-31-42-218:~$ cd ~
ubuntu@ip-172-31-42-218:~$ ls
My MyProject Project
ubuntu@ip-172-31-42-218:~$ cd MyProject/
ubuntu@ip-172-31-42-218:~/MyProject$ ls
MyProject
ubuntu@ip-172-31-42-218:~/MyProject$ cd MyProject/
ubuntu@ip-172-31-42-218:~/MyProject/MyProject$ git add .
ubuntu@ip-172-31-42-218:~/MyProject/MyProject$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   .gitignore
ubuntu@ip-172-31-42-218:~/MyProject/MyProject$ git commit -m "MyProject"
[master 5b2abbc] MyProject
1 file changed, 3 insertions(+)
create mode 100644 .gitignore
ubuntu@ip-172-31-42-218:~/MyProject/MyProject$
```

```
drwxr-xr-x 13 root root 4096 Oct 25 21:53 var
ubuntu@ip-172-31-42-218:/$ cd ~
ubuntu@ip-172-31-42-218:~$ ls
My MyProject Project
ubuntu@ip-172-31-42-218:~$ cd ~
ubuntu@ip-172-31-42-218:~$ ls
My MyProject Project
ubuntu@ip-172-31-42-218:~$ cd MyProject/
ubuntu@ip-172-31-42-218:~/MyProject$ ls
MyProject
ubuntu@ip-172-31-42-218:~/MyProject$ cd MyProject/
ubuntu@ip-172-31-42-218:~/MyProject/MyProject$ git add .
ubuntu@ip-172-31-42-218:~/MyProject/MyProject$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   .gitignore
ubuntu@ip-172-31-42-218:~/MyProject/MyProject$ git commit -m "MyProject"
[master 5b2abbc] MyProject
1 file changed, 3 insertions(+)
create mode 100644 .gitignore
ubuntu@ip-172-31-42-218:~/MyProject/MyProject$
```

github.com/new

Owner * somajoshi-git / Repository name * projectrepo
projectrepo is available.

Great repository names are short and memorable. Need inspiration? How about [miniature-computing-machine](#)?

Description (optional)

☒ Public
Anyone on the internet can see this repository. You choose who can commit.

☐ Private
You choose who can see and commit to this repository.

Initialize this repository with:

☐ Add a README file
This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore
.gitignore template: None

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license
License: None

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

github.com/somajoshi-git/projectrepo

somajoshi-git / projectrepo

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

projectrepo Public

Pin Unwatch 1 Fork 0 Star 0

Set up GitHub Copilot
Use GitHub's AI pair programmer to autocomplete suggestions as you code.
[Get started with GitHub Copilot](#)

Add collaborators to this repository
Search for people using their GitHub username or email address.
[Invite collaborators](#)

Quick setup — if you've done this kind of thing before

[Set up in Desktop](#) or [HTTPS](#) [SSH](#) <https://github.com/somajoshi-git/projectrepo.git>

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# projectrepo" >> README.md
git init
git add README.md
```

```

branch master
Changes to be committed:
  use "git restore --staged <file>..." to unstage)
    new file:   .gitignore

ubuntu@ip-172-31-42-218:~/MyProject/MyProject$ git commit -m "MyProject"
[master 5b2abbc] MyProject
1 file changed, 3 insertions(+)
create mode 100644 .gitignore
ubuntu@ip-172-31-42-218:~/MyProject/MyProject$ git remote add origin https://github.com/somajoshi-git/projectrepo.git
ubuntu@ip-172-31-42-218:~/MyProject/MyProject$

```

Get more information on the professional edition from <https://marketplace.visualstudio.com>

```

and the repository exists.
ubuntu@ip-172-31-42-218:~/MyProject/MyProject$ git remote -v
origin https://github.com/somajoshi-git/projectrepo.git (fetch)
origin https://github.com/somajoshi-git/projectrepo.git (push)
ubuntu@ip-172-31-42-218:~/MyProject/MyProject$

```

```

samacse.joshi@gmail.com: Command not found
ubuntu@ip-172-31-42-218:~/MyProject/MyProject$ git push -u origin master
Username for 'https://github.com': somacse.joshi@gmail.com
Password for 'https://samacse.joshi@gmail.com@github.com':
Enumerating objects: 53, done.
Counting objects: 100% (53/53), done.
Compressing objects: 100% (31/31), done.
Writing objects: 100% (53/53), 8.53 KiB | 1.22 MiB/s, done.
Total 53 (delta 2), reused 0 (delta 0)
Remote: Resolving deltas: 100% (2/2), done.
To https://github.com/somajoshi-git/projectrepo.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
ubuntu@ip-172-31-42-218:~/MyProject/MyProject$

```



somajoshi-git / projectrepo

<> Code

Issues

Pull requests

Actions

Projects

Wiki

Sec



projectrepo

Public

master

1 Branch 0 Tags

Go to file



somajoshi-git MyProject



.settings

initial commit



src

initial commit



target

initial commit



.classpath

ini



.gitignore

MyProject



.project

initial commit



pom.xml

initial commit



README



Type to search



ErrorSecurityGroupDevOps_TrainSA2310021GIT Assignprojectrepo/About remotGoogle Pass

github.com/somajoshi-git/projectrepo/blob/master/.gitignore

somajoshi-git / projectrepo

Type to search

<> Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

Files

master

Go to file

> .settings

> src

> main/java/com/assignment/my...

> test/java/com/assignment/mya...

> target

.classpath

.gitignore

.project

pom.xml

projectrepo / .gitignore

somajoshi-git MyProject

5b2abbc · 31 minutes agoHistory

CodeBlame

3 lines (3 loc) · 29 Bytes

Code 55% faster with GitHub Copilot

RawDownloadEdit

```
1 *.class
2 *.settings
3 *.project
```

Type to search

ENG IN

2:17 AM 1/5/2024

Assignment -2

```
Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT
$ mkdir javadir

Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT
$ cd javadir/

Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/javadir
$ git init
Initialized empty Git repository in D:/DevOpsGIT/javadir/.git/

Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/javadir (master)
$ touch file1.java

Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/javadir (master)
$ git add .

Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/javadir (master)
$ git commit -m"file1added"
[master (root-commit) eb59bb3] file1added
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 file1.java

Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/javadir (master)
$ touch file2.java

Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/javadir (master)
$ git add .

Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/javadir (master)
$ git commit -m"file2added"
[master 160382a] file2added
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 file2.java

Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/javadir (master)
$ git status
On branch master
nothing to commit, working tree clean

Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/javadir (master)
$ |
```

```
Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/javadir (master)
$ git status
On branch master
nothing to commit, working tree clean

Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/javadir (master)
$ touch file3.java

Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/javadir (master)
$ git add .

Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/javadir (master)
$ git stash save "file3"
Saved working directory and index state On master: file3

Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/javadir (master)
$ git stash list
stash@{0}: On master: file3

Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/javadir (master)
$ git stash apply stash@{0}
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   file3.java
```

```
Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/javadir (master)
$ git stash clear

Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/javadir (master)
$ git stash list

Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/javadir (master)
$
```

```
Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/javadir (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   file3.java
```

```
Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/javadir (master)
$ git add .
```

```
Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/javadir (master)
$ git commit -m"file3added"
[master 2320912] file3added
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 file3.java
```

```
Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/javadir (master)
$ git log --oneline
2320912 (HEAD -> master) file3added
160382a file2added
eb59bb3 file1added
```

```
Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/javadir (master)
```

```
Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/javadir (master)
$ git log --oneline
2320912 (HEAD -> master) file3added
160382a file2added
eb59bb3 file1added
```

```
Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/javadir (master)
$ git reset --soft eb59bb3 file1added
fatal: Cannot do soft reset with paths.
```

```
Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/javadir (master)
$ git reset --soft eb59bb3
```

```
Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/javadir (master)
$ ls
file1.java  file2.java  file3.java
```

```
Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/javadir (master)
$ git ls-l
git: 'ls-l' is not a git command. See 'git --help'.
```

```
The most similar command is
  ls-files
```

```
Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/javadir (master)
$ git ls-files
file1.java
file2.java
file3.java
```

```
Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/javadir (master)
$
```

```
MINGW64:/d/DevOpsGIT/javadir
user@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/javadir (master)
$ git log --one
fatal: unrecognized argument: --one

user@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/javadir (master)
$ git log --oneline
eb59bb3 (HEAD -> master) file1added

user@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/javadir (master)
$ cd ..

user@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT
$ cd javadir

user@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/javadir (master)
$ ls
file1.java file2.java file3.java

user@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/javadir (master)
$ git revert eb59bb3
error: your local changes would be overwritten by revert.
hint: commit your changes or stash them to proceed.
fatal: revert failed

user@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/javadir (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   file2.java
        new file:   file3.java

user@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/javadir (master)
$ git add .

user@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/javadir (master)
$ git commit -m"added"
[master 3c0f89b] added
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 file2.java
 create mode 100644 file3.java

user@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/javadir (master)
$ git status
On branch master
nothing to commit, working tree clean

user@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/javadir (master)
$ git log --oneline
3c0f89b (HEAD -> master) added
eb59bb3 file1added

user@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/javadir (master)
$
```

```
added

user@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/javadir (master)
$ git reset --mixed eb59bb3

user@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/javadir (master)
$ ls
file1.java file2.java file3.java

user@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/javadir (master)
$ git ls-files
file1.java

user@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/javadir (master)
```



```
Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/javadir (master)
$ ls
file1.java  file2.java  file3.java

Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/javadir (master)
$ git ls-files
file1.java

Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/javadir (master)
$ git log --oneline
eb59bb3 (HEAD -> master) file1added

Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/javadir (master)
$ git revert eb59bb3
[master 0f9ba72] Revert "file1added"
 1 file changed, 0 insertions(+), 0 deletions(-)
 delete mode 100644 file1.java

Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/javadir (master)
$ |
```

```
Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/javadir (master)
$ git add .

Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/javadir (master)
$ git commit -m"files added"
[master 04146bc] files added
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 file2.java
 create mode 100644 file3.java

Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/javadir (master)
$
```

```
Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/javadir (master)
$ git reset --hard
HEAD is now at 04146bc files added
```

Assignment-3

```
Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/javadir (master)
$ echo "hello world" >>filedemo.java

Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/javadir (master)
$ git branch
* master

Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/javadir (master)
$ cat filedemo
cat: filedemo: No such file or directory

Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/javadir (master)
$ ls
file2.java  file3.java  filedemo.java

Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/javadir (master)
$ cat filedemo.java
hello world

Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/javadir (master)
$ |
```

git

branch

```
Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/new (master)
$ echo "hello world" >>filedemo.java

Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/new (master)
$ ls
1.txt 2.txt e.txt f.txt filedemo.java g.txt j.txt r.txt s.txt u.txt

Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/new (master)
$ git add .
warning: in the working copy of 'filedemo.java', LF will be replaced by CRLF the next time Git touches it

Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/new (master)
$ git commit -m"filedemo.java"
[master d5cfe42] filedemo.java
1 file changed, 1 insertion(+)
 create mode 100644 filedemo.java

Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/new (master)
$ git status
On branch master
nothing to commit, working tree clean

Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/new (master)
$
```

```
Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT
$ mkdir demo

Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT
$ cd demo

Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/demo
$ git init
Initialized empty Git repository in D:/DevOpsGIT/demo/.git/

Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/demo (master)
$ echo "hello world" >>Demo.java

Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/demo (master)
$ cat Demo.java
hello world

Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/demo (master)
$ git add .
warning: in the working copy of 'Demo.java', LF will be replaced by CRLF the next time Git touches it
g
Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/demo (master)
$ git commit -m"Demo.java"
[master (root-commit) 1432440] Demo.java
1 file changed, 1 insertion(+)
create mode 100644 Demo.java

Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/demo (master)
$ |
```

```
Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/demo (master)
$ git branch -c feature1

Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/demo (master)
$ git status
On branch master
nothing to commit, working tree clean

Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/demo (master)
$ git switch feature1
Switched to branch 'feature1'

Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/demo (feature1)
$ ls
Demo.java

Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/demo (feature1)
$ git switch master
Switched to branch 'master'

Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/demo (master)
$ git branch -c feature2

Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/demo (master)
$ git switch feature2
Switched to branch 'feature2'

Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/demo (feature2)
$ ls
Demo.java

Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/demo (feature2)
```

```
MINGW64:/d/DevOpsGIT/demo
user@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/demo (feature2)
$ ls
demo.java

user@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/demo (feature2)
$ echo "rec1" >>file1

user@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/demo (feature2)
$ git add
warning: in the working copy of 'file1', LF will be replaced by CRLF the next time Git touches it

user@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/demo (feature2)
$ git commit -m"rec1file1"
[feature2 cfcf4ae] rec1file1
1 file changed, 1 insertion(+)
create mode 100644 file1

user@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/demo (feature2)
$ git switch master
Switched to branch 'master'

user@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/demo (master)
$ git merge feature2
Updating 1432440..cfcf4ae
Fast-forward
1 file changed, 1 insertion(+)
create mode 100644 file1

user@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/demo (master)
$ git switch feature1
Switched to branch 'feature1'

user@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/demo (feature1)
$ echo "rec2" >>file1

user@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/demo (feature1)
$ git add
warning: in the working copy of 'file1', LF will be replaced by CRLF the next time Git touches it

user@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/demo (feature1)
$ git commit -m"rec2file1"
[feature1 93b6363] rec2file1
1 file changed, 1 insertion(+)
create mode 100644 file1

user@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/demo (feature1)
$ git switch master
Switched to branch 'master'

user@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/demo (master)
$ git merge feature1
Auto-merging file1
CONFLICT (add/add): Merge conflict in file1
Automatic merge failed; fix conflicts and then commit the result.

user@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/demo (master|MERGING)
$
```

```
user@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/demo (master|MERGING)
$ git mergetool

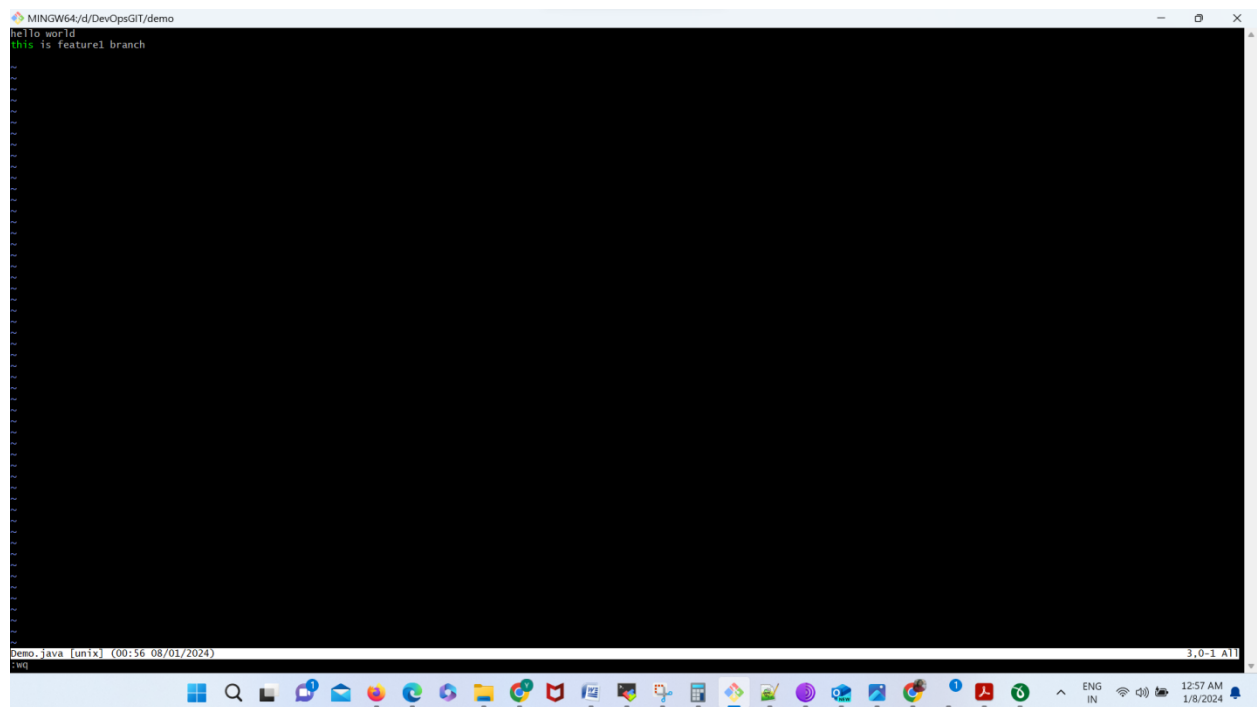
This message is displayed because 'merge.tool' is not configured.
See 'git mergetool --tool-help' or 'git help config' for more details.
git mergetool will now attempt to use one of the following tools:
pdiff kdiff3 tkdiff xxdiff meld tortoisemerge gvimdiff diffmerge ecmerge p4merge araxis bc codecompare smerge emerge vimdiff nvimdiff
merging:
file1

Normal merge conflict for 'file1':
{local}: created file
{remote}: created file
git return to start merge resolution tool (vimdiff):
3 files to edit

user@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/demo (master|MERGING)
```

```
MINGW64/d/DevOpsGIT/demo
Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/demo (feature1)
$ git add .
warning: in the working copy of 'file2', LF will be replaced by CRLF the next time Git touches it
Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/demo (feature1)
$ git commit -m"recafile2"
[feature1 aa2143f] recafile2
1 file changed, 1 insertion(+)
create mode 100644 file2
Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/demo (feature1)
$ git switch master
Switched to branch 'master'
Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/demo (master)
$ git switch feature2
Switched to branch 'feature2'
Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/demo (feature2)
$ echo recB>>file2
Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/demo (feature2)
$ git add .
warning: in the working copy of 'file2', LF will be replaced by CRLF the next time Git touches it
Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/demo (feature2)
$ git commit -m"recbfile2"
[feature2 1bdec9] recbfile2
1 file changed, 1 insertion(+)
create mode 100644 file2
Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/demo (feature2)
$ git switch master
Switched to branch 'master'
Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/demo (master)
$ git merge feature2
Merge made by the 'ort' strategy.
file2 | 1 +
1 file changed, 1 insertion(+)
create mode 100644 file2
Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/demo (master)
$ git switch feature2
Switched to branch 'feature2'
Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/demo (feature2)
$ git switch master
Switched to branch 'master'
Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/demo (master)
$ git merge feature2
Auto-merging file2
CONFLICT (add/add): Merge conflict in file2
Automatic merge failed; fix conflicts and then commit the result.
Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/demo (master|MERGING)
$
```

```
Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/demo (feature1)
$ vi Demo.java
Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/demo (feature1)
$
```

```
Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/demo (feature2)
$ vi Demo.java

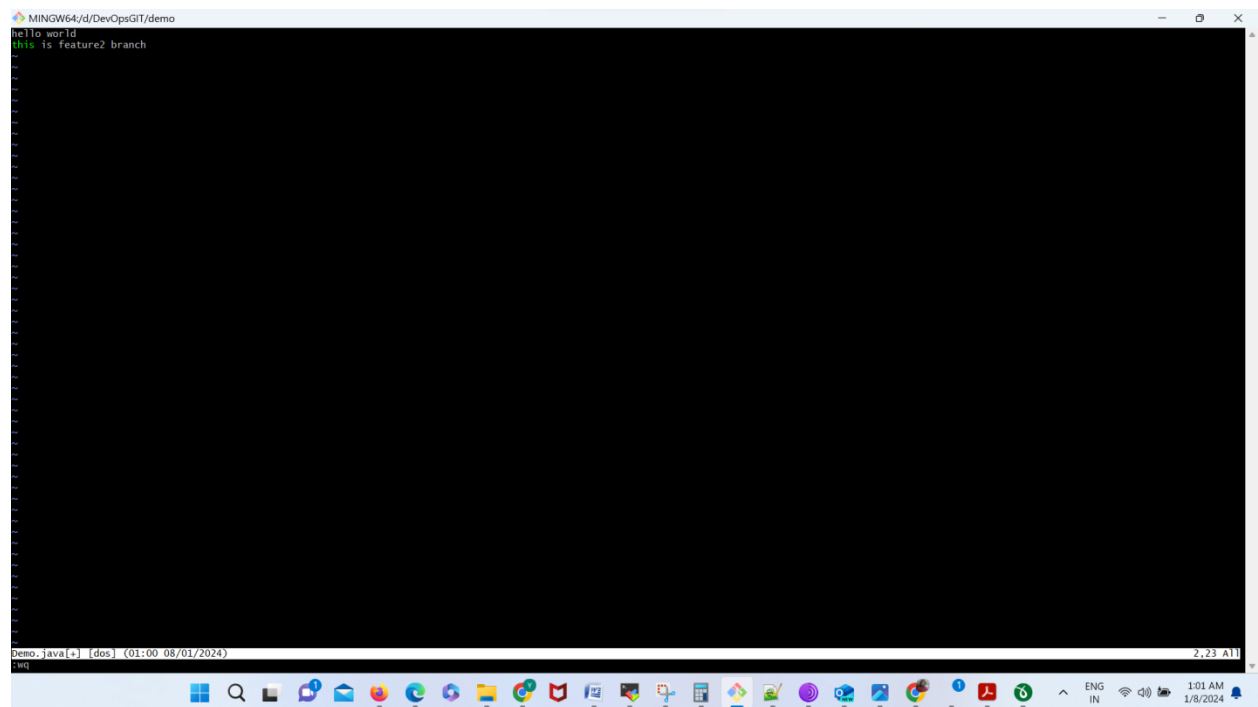
Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/demo (feature2)
```

```
Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/demo (feature2)
$ git switch master
Switched to branch 'master'

Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/demo (master)
$ git merge feature1
Merge made by the 'ort' strategy.
 Demo.java | 2 ++
 file2.orig | 5 +++++
 2 files changed, 7 insertions(+)
 create mode 100644 file2.orig

Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/demo (master)
$ git merge feature2
Auto-merging Demo.java
CONFLICT (content): Merge conflict in Demo.java
Auto-merging file2
CONFLICT (add/add): Merge conflict in file2
Automatic merge failed; fix conflicts and then commit the result.

Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/demo (master|MERGING)
$
```



```
hello world
<<<<<<< HEAD
this is feature1 branch

=====
this is feature2 branch
this is in merge
>>>>>> feature2
~
~
~
~
~
```

```
hello worldthis is feature1 branch
this is feature2 branch
this is in merge
```

```
Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/demo (master|MERGING)
$ vi Demo.java
```

```
Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/demo (master|MERGING)
$ git add .
```

```
Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/demo (master|MERGING)
$ git commit -m"resolve"
[master b7cb085] resolve
```

```
Acer@DESKTOP-PV788EP MINGW64 /d/DevOpsGIT/demo (master)
```

```
$ git log --oneline
b7cb085 (HEAD -> master) resolve
ale91e2 (feature2) df2
d29b167 Merge branch 'feature1'
395735d Demof2
49d7b36 Demo.java
d4e98c0 Merge branch 'feature1'
1bdedc9 recbfile2
aa2143f recafile2
a43b4fc bjh
93b6363 rec2file1
cfcf4ae rec1file1
1432440 Demo.java
```