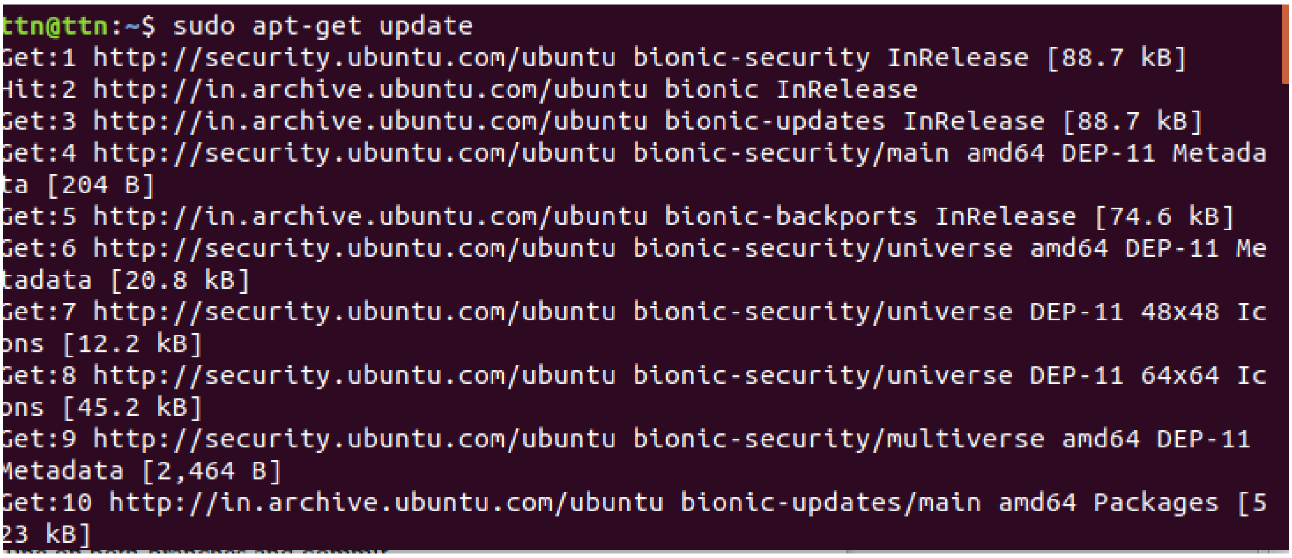
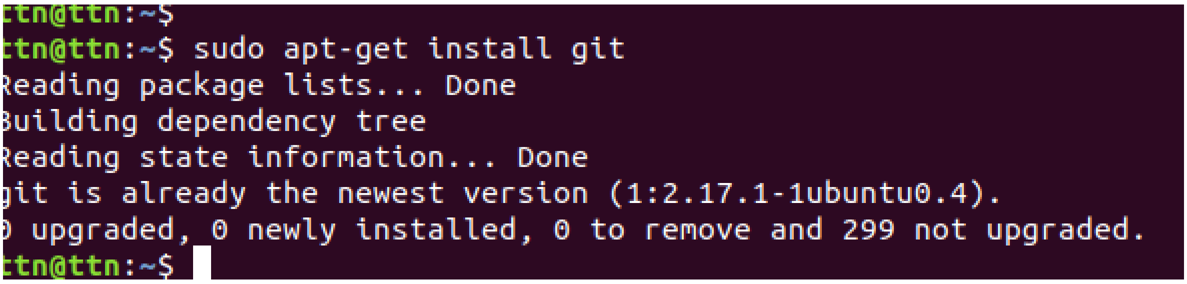
Version Control and Git Assignment

Q1. Git Setup https://confluence.atlassian.com/bitbucket/set-up-git-744723531.html

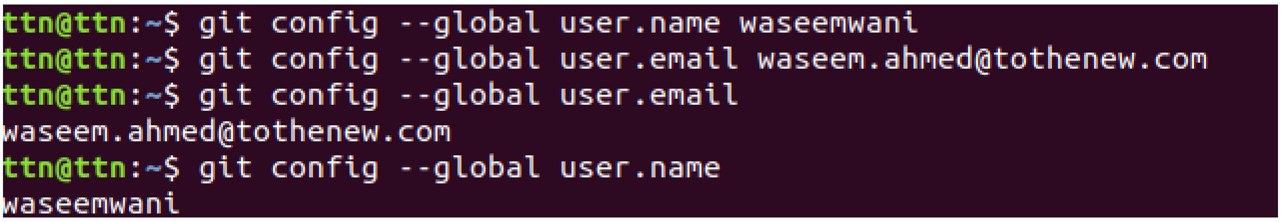
Ans.



```
ttn@ttn:~$ sudo apt-get update
Get:1 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Hit:2 http://in.archive.ubuntu.com/ubuntu bionic InRelease
Get:3 http://in.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:4 http://security.ubuntu.com/ubuntu bionic-security/main amd64 DEP-11 Metada
ta [204 B]
Get:5 http://in.archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Get:6 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 DEP-11 Me
tadata [20.8 kB]
Get:7 http://security.ubuntu.com/ubuntu bionic-security/universe DEP-11 48x48 Ic
ons [12.2 kB]
Get:8 http://security.ubuntu.com/ubuntu bionic-security/universe DEP-11 64x64 Ic
ons [45.2 kB]
Get:9 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 DEP-11
Metadata [2,464 B]
Get:10 http://in.archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages [5
23 kB]
```

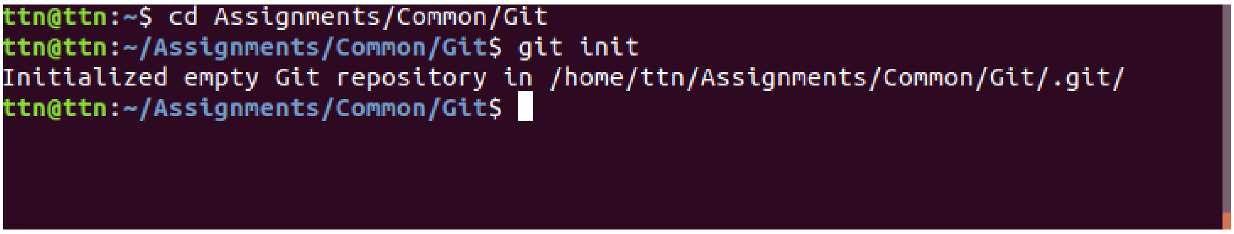


```
ttn@ttn:~$
ttn@ttn:~$ sudo apt-get install git
Reading package lists... Done
Building dependency tree
Reading state information... Done
git is already the newest version (1:2.17.1-1ubuntu0.4).
0 upgraded, 0 newly installed, 0 to remove and 299 not upgraded.
ttn@ttn:~$
```



```
ttn@ttn:~$ git config --global user.name waseemwani
ttn@ttn:~$ git config --global user.email waseem.ahmed@tothenew.com
ttn@ttn:~$ git config --global user.email
waseem.ahmed@tothenew.com
ttn@ttn:~$ git config --global user.name
waseemwani
```

- **Q2.** Initialize a Git Repository



```
ttn@ttn:~$ cd Assignments/Common/Git
ttn@ttn:~/Assignments/Common/Git$ git init
Initialized empty Git repository in /home/ttn/Assignments/Common/Git/.git/
ttn@ttn:~/Assignments/Common/Git$
```

Q3. Add files to the repository
Ans. first create a file as

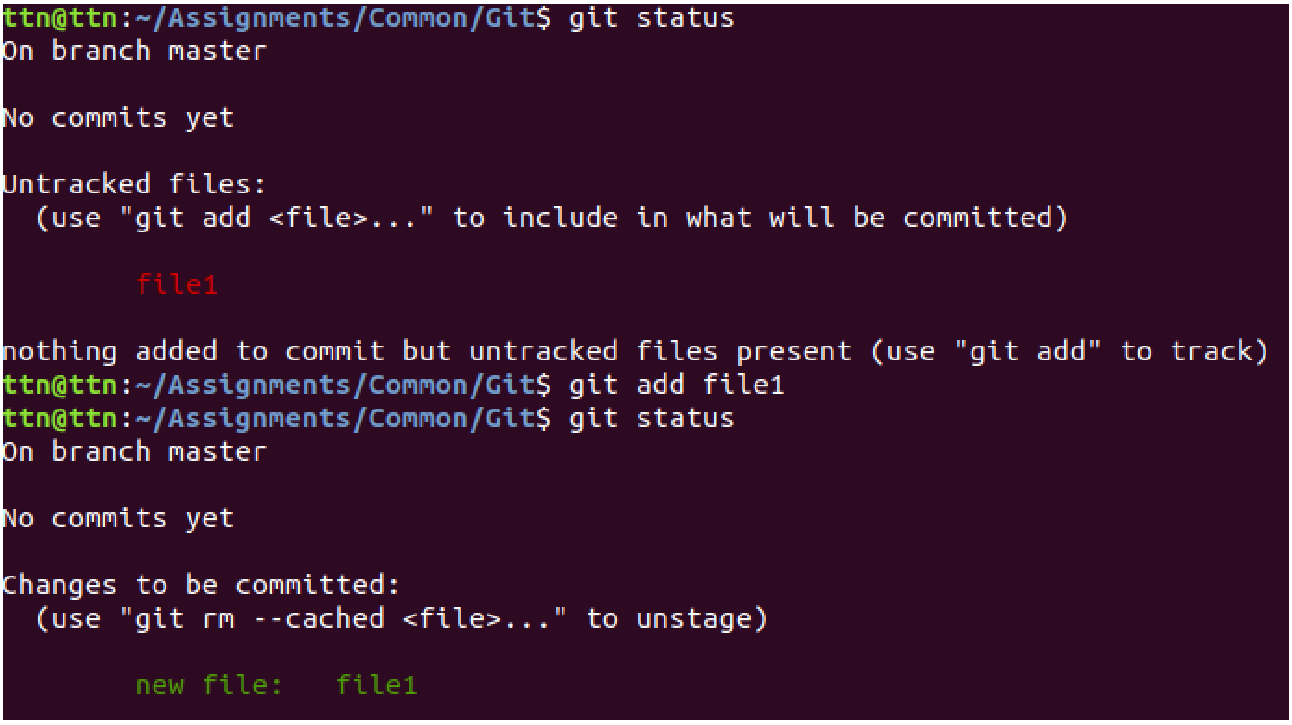
$ touch file1

then add it to the repository as

$ git add file1

then check status

Demonstration as shown below



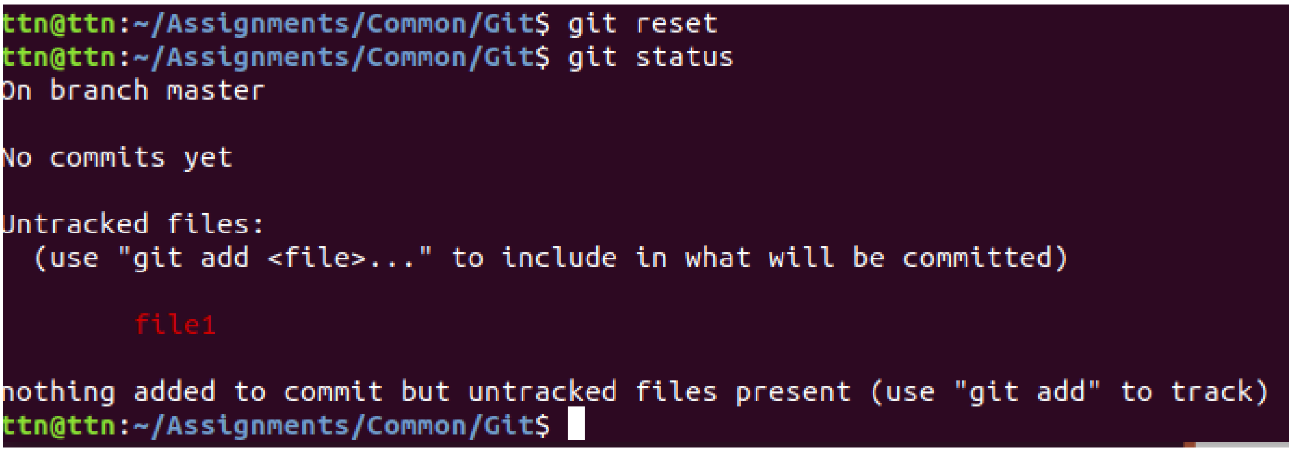
```
ttn@ttn:~/Assignments/Common/Git$ git status
On branch master

No commits yet

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

        file1

nothing added to commit but untracked files present (use "git add" to track)
ttn@ttn:~/Assignments/Common/Git$ git add file1
ttn@ttn:~/Assignments/Common/Git$ git status
On branch master

No commits yet

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

        new file:   file1
```

Q4. Unstage 1 file

a file can be unstaged from the index using the commands....

$ git reset file -- filename

or $ git rm –cached filename

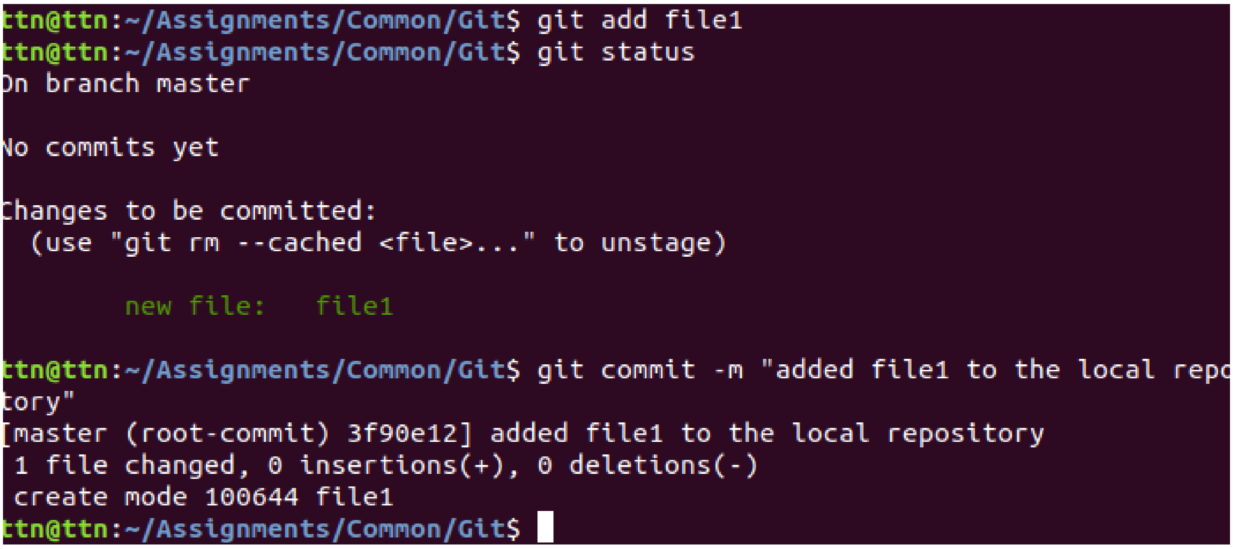


```
ttn@ttn:~/Assignments/Common/Git$ git reset
ttn@ttn:~/Assignments/Common/Git$ git status
On branch master

No commits yet

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

        file1

nothing added to commit but untracked files present (use "git add" to track)
ttn@ttn:~/Assignments/Common/Git$ 
```

Q5. Commit the file

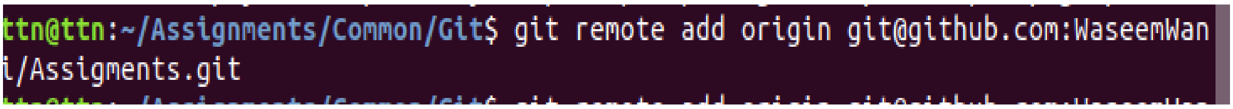
Ans. first stage the file in the index
then commit using the command
$git commit -m “ commit message”

Demonstarttion in the below image



```
ttn@ttn:~/Assignments/Common/Git$ git add file1
ttn@ttn:~/Assignments/Common/Git$ git status
On branch master

No commits yet

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

        new file:   file1

ttn@ttn:~/Assignments/Common/Git$ git commit -m "added file1 to the local repo
tory"
[master (root-commit) 3f90e12] added file1 to the local repository
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 file1
ttn@ttn:~/Assignments/Common/Git$ 
```

Q6. Add a remote

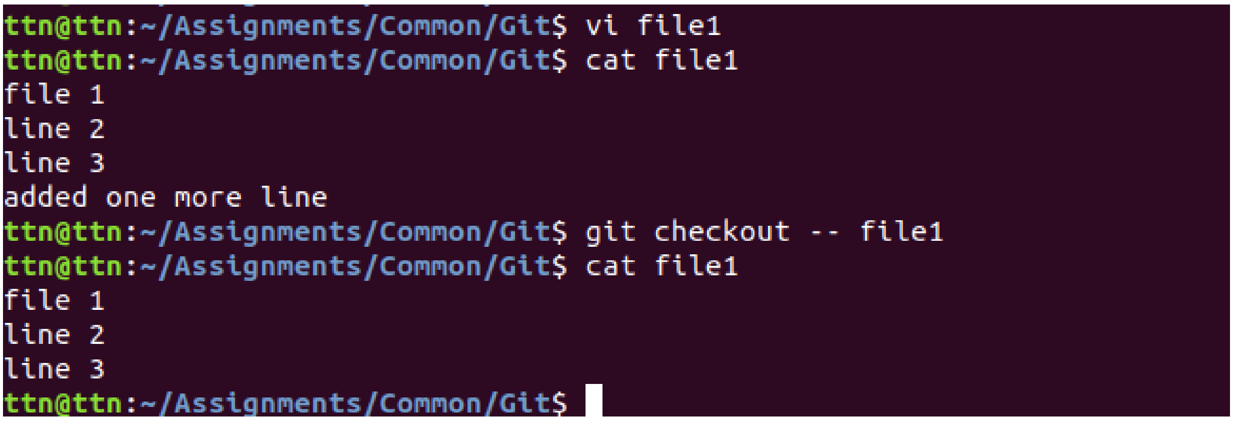


```
ttn@ttn:~/Assignments/Common/Git$ git remote add origin git@github.com:WaseemWan
i/Assigments.git
```

Q7. Undo changes to a particular file

Ans. First modified the file, file1 by adding some more content
then undone the changes using the comand
$ git checkout -- file1

Demonstration shown in below image



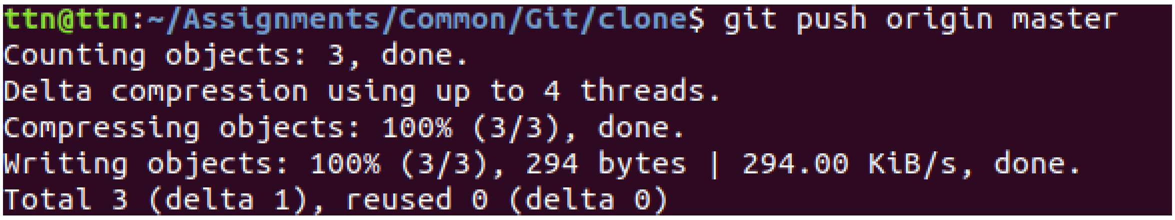
```
ttn@ttn:~/Assignments/Common/Git$ vi file1
ttn@ttn:~/Assignments/Common/Git$ cat file1
file 1
line 2
line 3
added one more line
ttn@ttn:~/Assignments/Common/Git$ git checkout -- file1
ttn@ttn:~/Assignments/Common/Git$ cat file1
file 1
line 2
line 3
ttn@ttn:~/Assignments/Common/Git$
```

Q8. Push changes to Github

Ans. Changes can be pushed to git using the following command

$git push origin master

where origin means the destination for pushing the changes

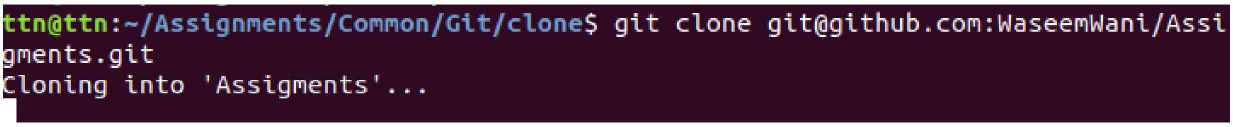


```
ttn@ttn:~/Assignments/Common/Git/clone$ git push origin master
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 294 bytes | 294.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0)
```

Q9. Clone the repository

Ans. First link of the repository is to be copied from the github and on the terminal , it can be cloned as

$git clone git@github.com:WaseemWani/Assigments.git



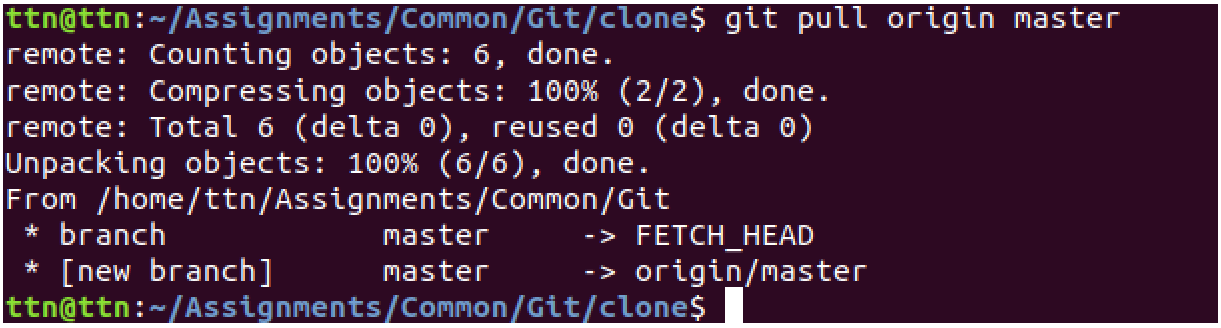
```
ttn@ttn:~/Assignments/Common/Git/clone$ git clone git@github.com:WaseemWani/Assi
gments.git
Cloning into 'Assigments'...
```

Q10. Add changes to one of the copies and pull the changes in the other.

Ans. Create a new folder using command

$ mkdir new1

now pull the changes using the command



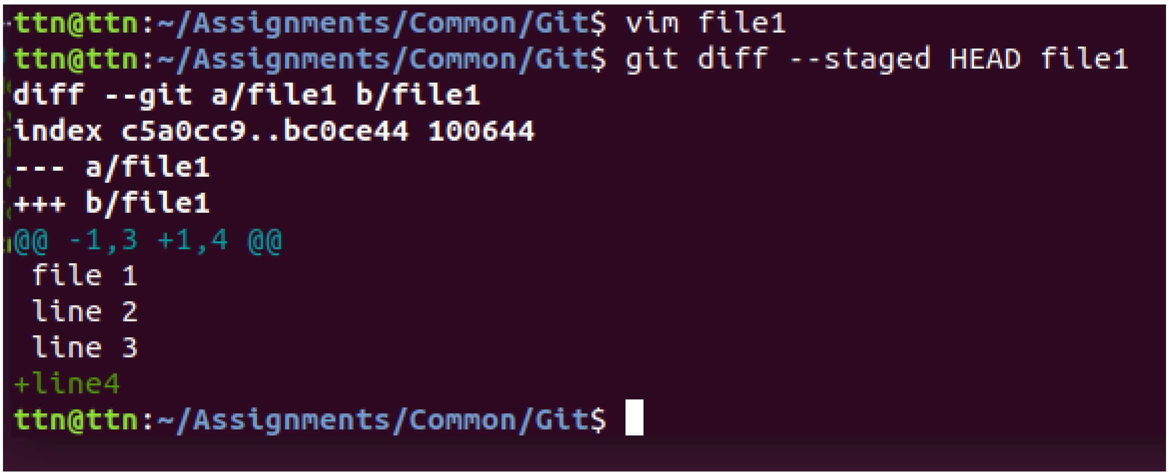
```
ttn@ttn:~/Assignments/Common/Git/clone$ git pull origin master
remote: Counting objects: 6, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 6 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (6/6), done.
From /home/ttn/Assignments/Common/Git
 * branch            master     -> FETCH_HEAD
 * [new branch]      master     -> origin/master
ttn@ttn:~/Assignments/Common/Git/clone$
```

Q11. Check differences between a file and its staged version

Ans. First edited a file named file1, which was staged earlier.

Then found the difference between changed file and its staged version using the command as shown below

$git diff –staged HEAD file1

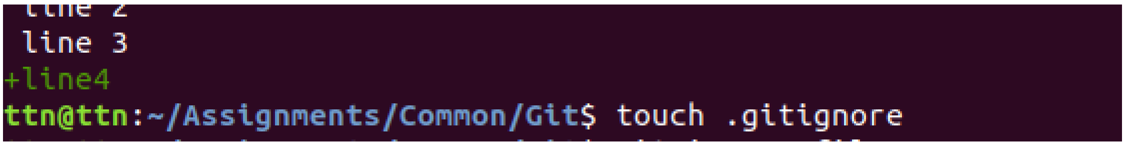


```
ttn@ttn:~/Assignments/Common/Git$ vim file1
ttn@ttn:~/Assignments/Common/Git$ git diff --staged HEAD file1
diff --git a/file1 b/file1
index c5a0cc9..bc0ce44 100644
--- a/file1
+++ b/file1
@@ -1,3 +1,4 @@
 file 1
 line 2
 line 3
+line4
ttn@ttn:~/Assignments/Common/Git$ 
```

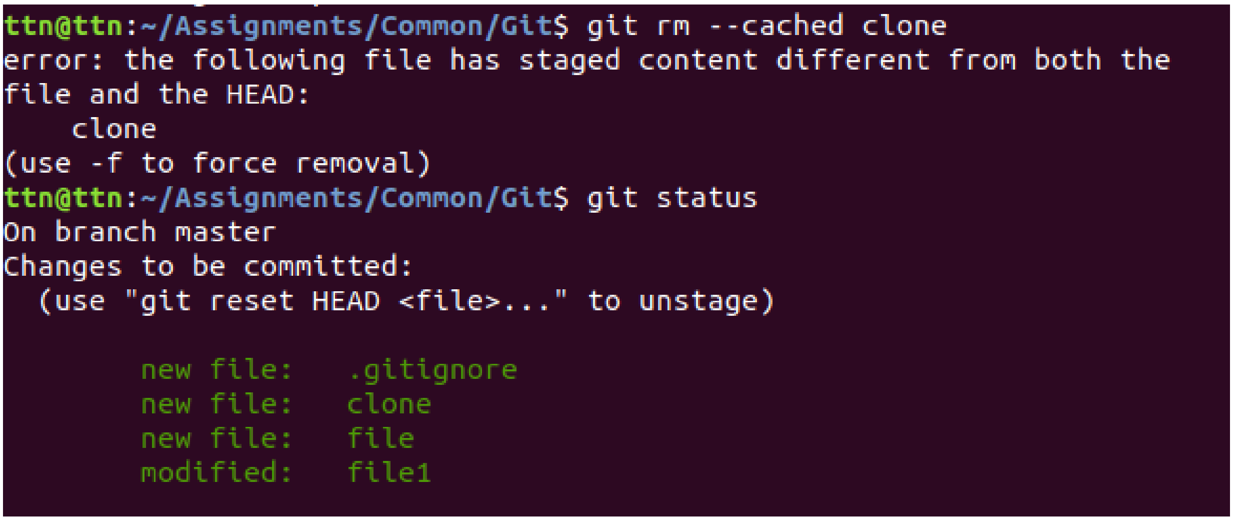
Q12. Ignore a few files to be checked in

Ans. Fisrt created a file .gitignore. Then added some patterns in it to ignore the files matching the pattern. Then created a file with the stored pattern externsion to check the effect of .gitignore.

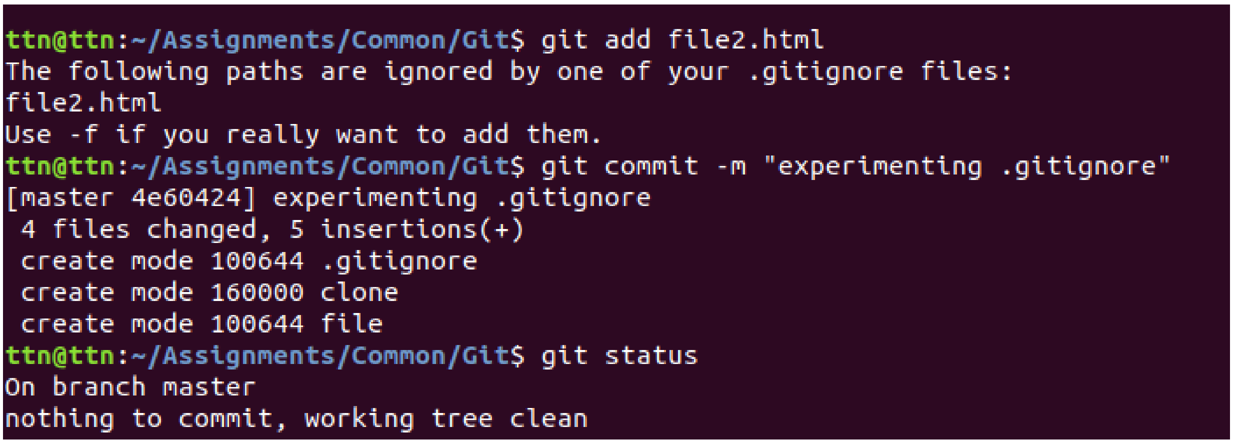
Demonstration in below images.



```
 line 2
 line 3
+line4
ttn@ttn:~/Assignments/Common/Git$ touch .gitignore
```

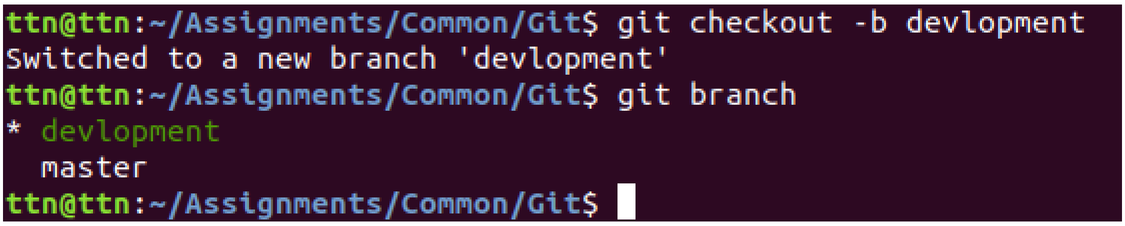


```
ttn@ttn:~/Assignments/Common/Git$ git rm --cached clone
error: the following file has staged content different from both the
file and the HEAD:
    clone
(use -f to force removal)
ttn@ttn:~/Assignments/Common/Git$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

	new file:   .gitignore
	new file:   clone
	new file:   file
	modified:   file1
```



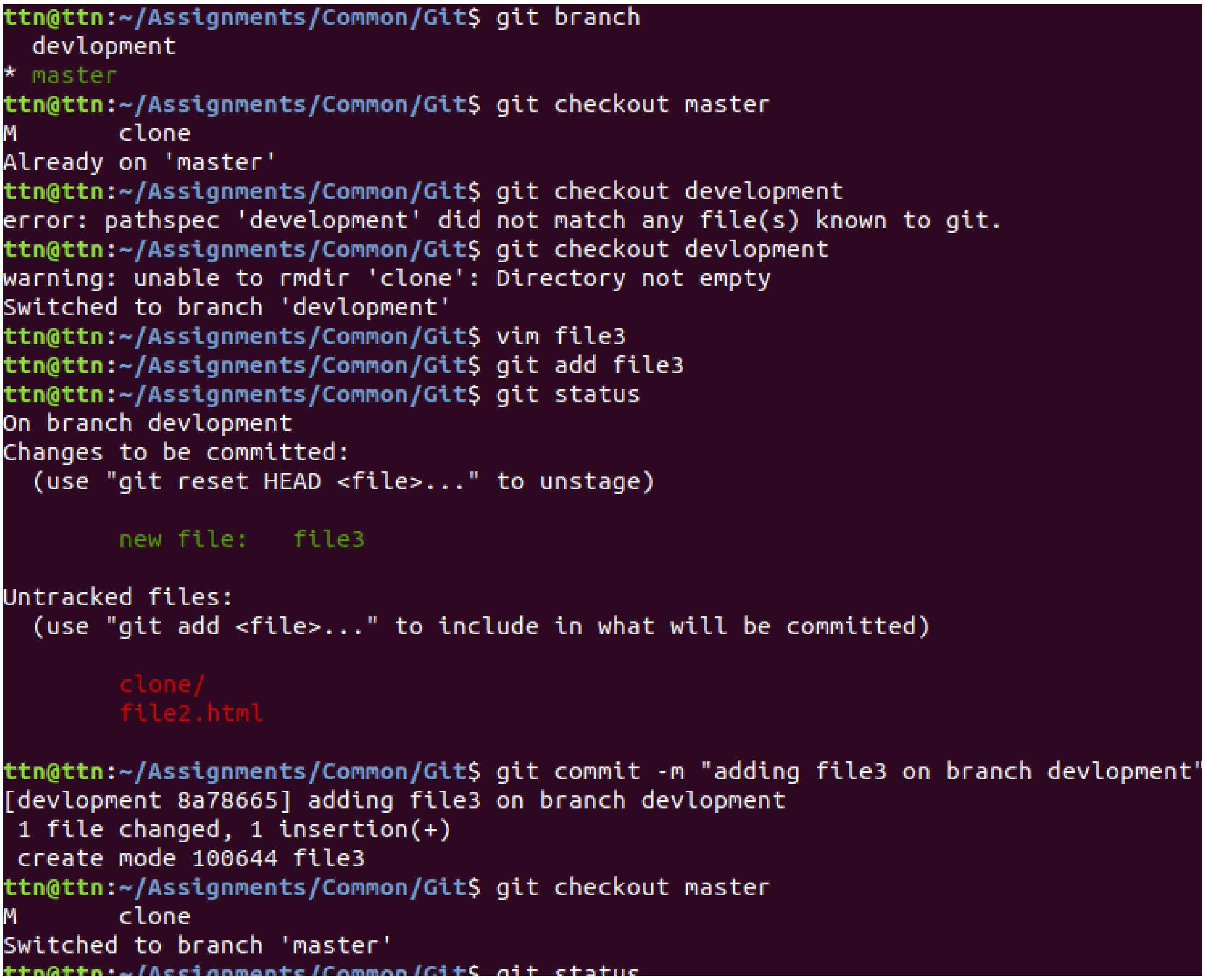
```
ttn@ttn:~/Assignments/Common/Git$ git add file2.html
The following paths are ignored by one of your .gitignore files:
file2.html
Use -f if you really want to add them.
ttn@ttn:~/Assignments/Common/Git$ git commit -m "experimenting .gitignore"
[master 4e60424] experimenting .gitignore
 4 files changed, 5 insertions(+)
 create mode 100644 .gitignore
 create mode 160000 clone
 create mode 100644 file
ttn@ttn:~/Assignments/Common/Git$ git status
On branch master
nothing to commit, working tree clean
```

Q13. Create a new branch.

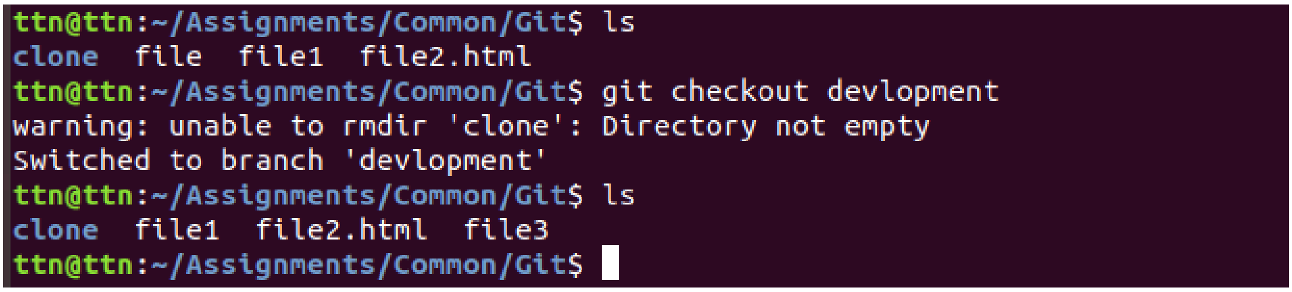


```
ttn@ttn:~/Assignments/Common/Git$ git checkout -b devlopment
Switched to a new branch 'devlopment'
ttn@ttn:~/Assignments/Common/Git$ git branch
* devlopment
  master
ttn@ttn:~/Assignments/Common/Git$ 
```

Q14. Diverge them with commits



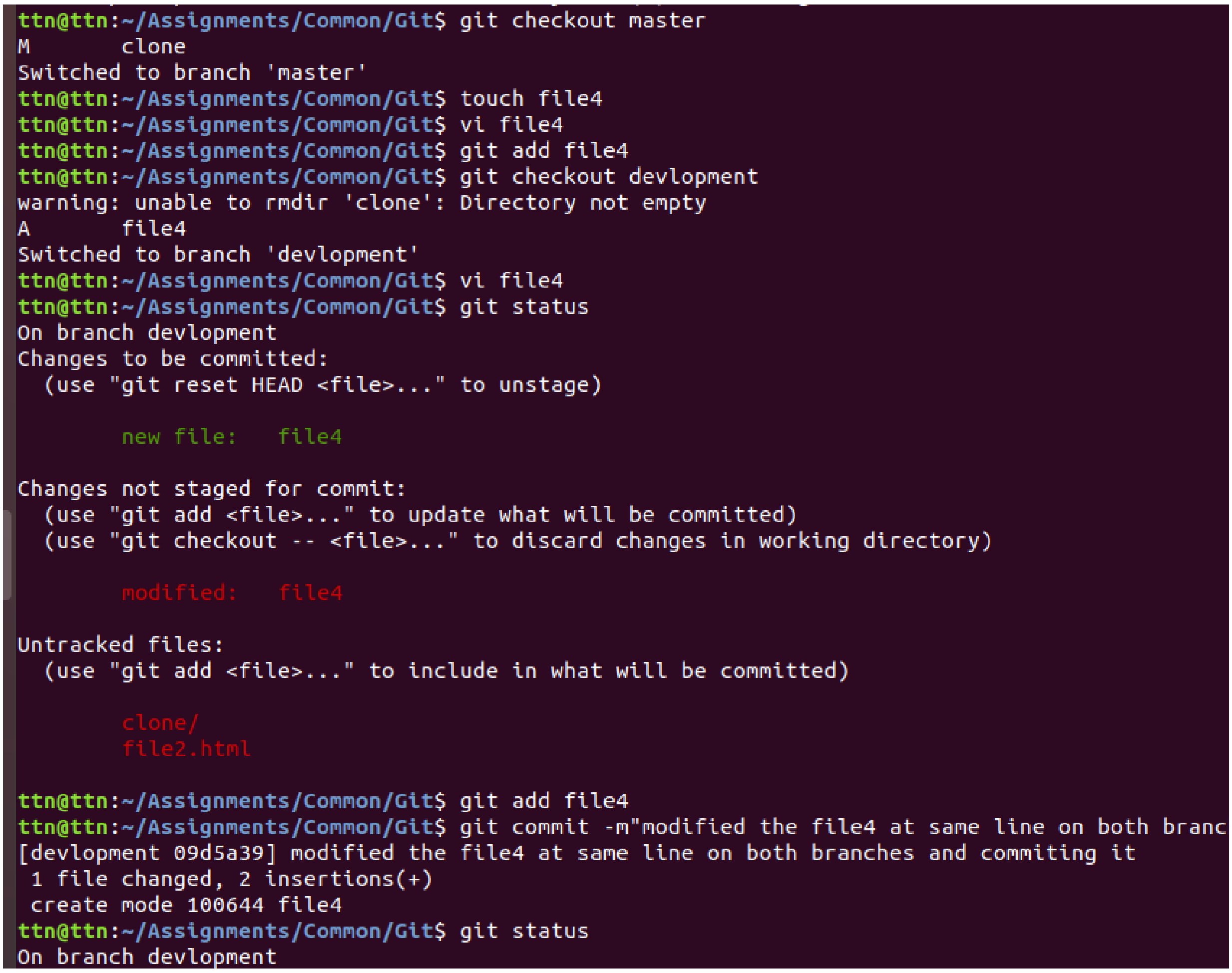
```
ttn@ttn:~/Assignments/Common/Git$ git branch
  devlopment
* master
ttn@ttn:~/Assignments/Common/Git$ git checkout master
M       clone
Already on 'master'
ttn@ttn:~/Assignments/Common/Git$ git checkout development
error: pathspec 'development' did not match any file(s) known to git.
ttn@ttn:~/Assignments/Common/Git$ git checkout devlopment
warning: unable to rmdir 'clone': Directory not empty
Switched to branch 'devlopment'
ttn@ttn:~/Assignments/Common/Git$ vim file3
ttn@ttn:~/Assignments/Common/Git$ git add file3
ttn@ttn:~/Assignments/Common/Git$ git status
On branch devlopment
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   file3

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

        clone/
        file2.html

ttn@ttn:~/Assignments/Common/Git$ git commit -m "adding file3 on branch devlopment"
[devlopment 8a78665] adding file3 on branch devlopment
 1 file changed, 1 insertion(+)
 create mode 100644 file3
ttn@ttn:~/Assignments/Common/Git$ git checkout master
M       clone
Switched to branch 'master'
ttn@ttn:~/Assignments/Common/Git$ git status
```



```
ttn@ttn:~/Assignments/Common/Git$ ls
clone  file  file1  file2.html
ttn@ttn:~/Assignments/Common/Git$ git checkout devlopment
warning: unable to rmdir 'clone': Directory not empty
Switched to branch 'devlopment'
ttn@ttn:~/Assignments/Common/Git$ ls
clone  file1  file2.html  file3
ttn@ttn:~/Assignments/Common/Git$ 
```

Q15. Edit the same file at the same line on both branches and commit

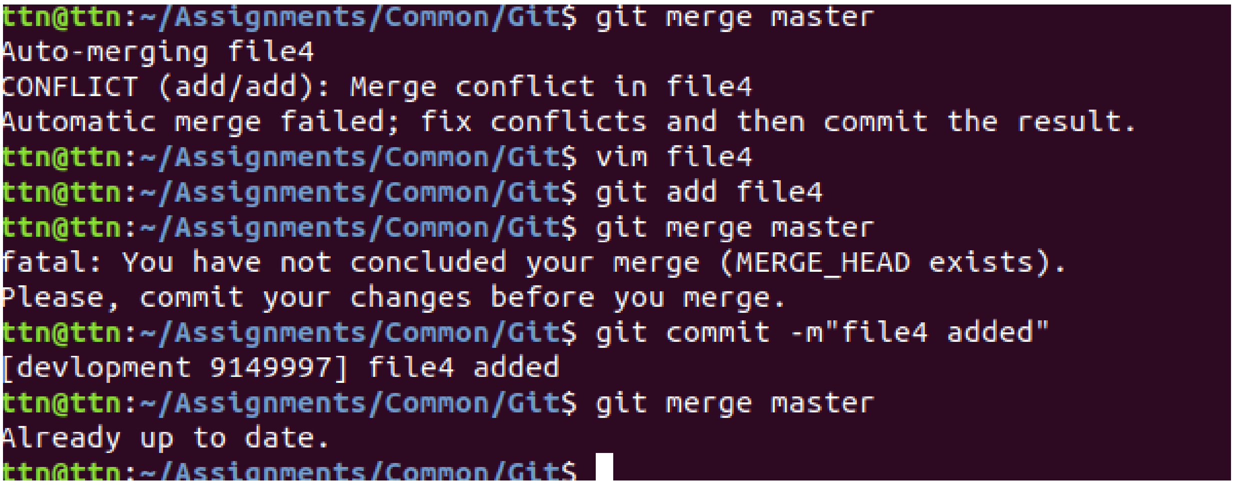
Ans.



```
ttn@ttn:~/Assignments/Common/Git$ git checkout master
M       clone
Switched to branch 'master'
ttn@ttn:~/Assignments/Common/Git$ touch file4
ttn@ttn:~/Assignments/Common/Git$ vi file4
ttn@ttn:~/Assignments/Common/Git$ git add file4
ttn@ttn:~/Assignments/Common/Git$ git checkout devlopment
warning: unable to rmdir 'clone': Directory not empty
A       file4
Switched to branch 'devlopment'
ttn@ttn:~/Assignments/Common/Git$ vi file4
ttn@ttn:~/Assignments/Common/Git$ git status
On branch devlopment
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   file4

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

        modified:   file4

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

        clone/
        file2.html

ttn@ttn:~/Assignments/Common/Git$ git add file4
ttn@ttn:~/Assignments/Common/Git$ git commit -m"modified the file4 at same line on both branc
[devlopment 09d5a39] modified the file4 at same line on both branches and commiting it
 1 file changed, 2 insertions(+)
 create mode 100644 file4
ttn@ttn:~/Assignments/Common/Git$ git status
On branch devlopment
```

Q16. Try merging and resolve merge conflicts

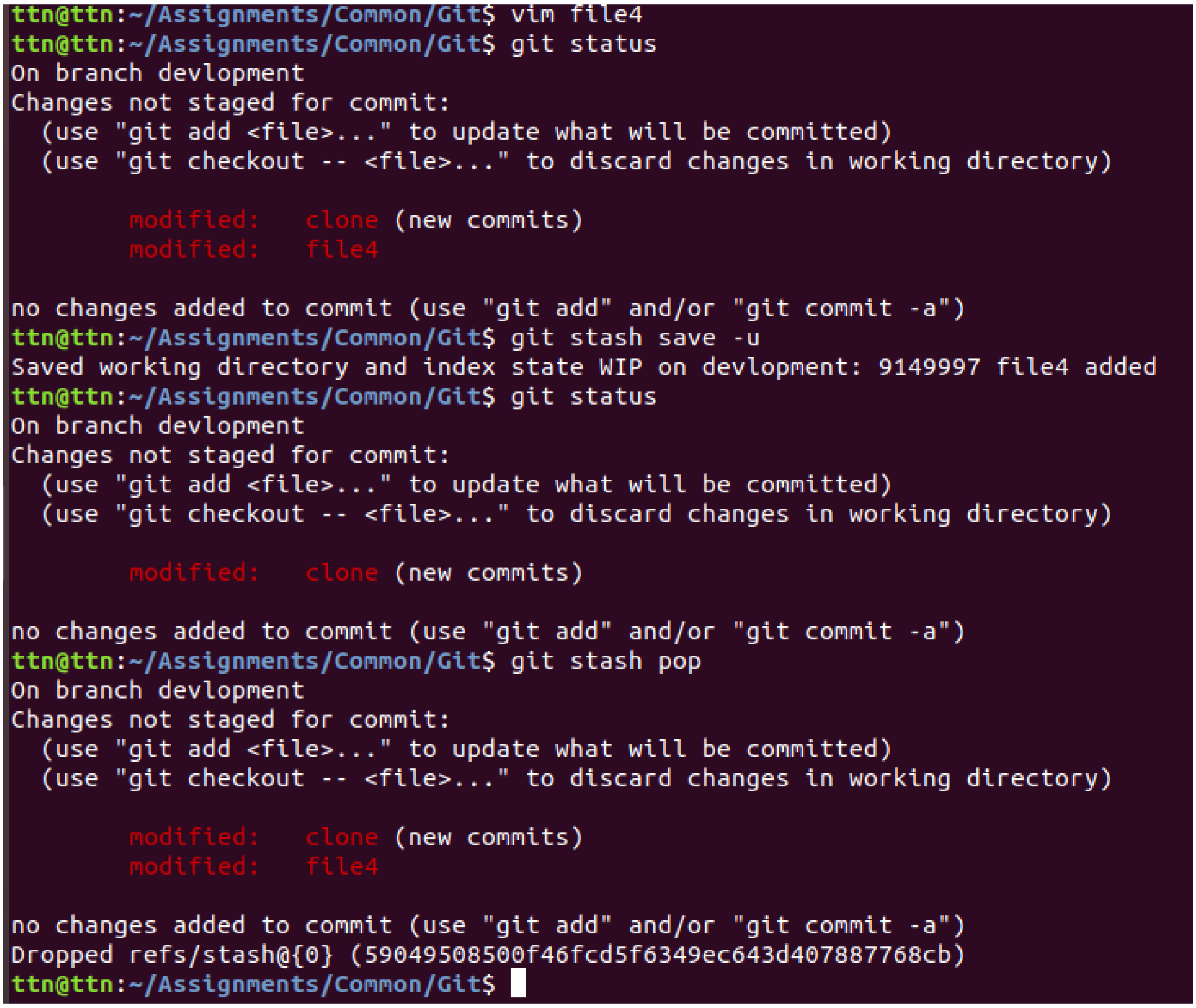
Ans.



```
ttn@ttn:~/Assignments/Common/Git$ git merge master
Auto-merging file4
CONFLICT (add/add): Merge conflict in file4
Automatic merge failed; fix conflicts and then commit the result.
ttn@ttn:~/Assignments/Common/Git$ vim file4
ttn@ttn:~/Assignments/Common/Git$ git add file4
ttn@ttn:~/Assignments/Common/Git$ git merge master
fatal: You have not concluded your merge (MERGE_HEAD exists).
Please, commit your changes before you merge.
ttn@ttn:~/Assignments/Common/Git$ git commit -m"file4 added"
[devlopment 9149997] file4 added
ttn@ttn:~/Assignments/Common/Git$ git merge master
Already up to date.
ttn@ttn:~/Assignments/Common/Git$
```

Q17. Stash the changes and pop them

Ans.



```
ttn@ttn:~/Assignments/Common/Git$ vim file4
ttn@ttn:~/Assignments/Common/Git$ git status
On branch devlopment
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   clone (new commits)
        modified:   file4

no changes added to commit (use "git add" and/or "git commit -a")
ttn@ttn:~/Assignments/Common/Git$ git stash save -u
Saved working directory and index state WIP on devlopment: 9149997 file4 added
ttn@ttn:~/Assignments/Common/Git$ git status
On branch devlopment
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   clone (new commits)

no changes added to commit (use "git add" and/or "git commit -a")
ttn@ttn:~/Assignments/Common/Git$ git stash pop
On branch devlopment
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   clone (new commits)
        modified:   file4

no changes added to commit (use "git add" and/or "git commit -a")
Dropped refs/stash@{0} (59049508500f46fcd5f6349ec643d407887768cb)
ttn@ttn:~/Assignments/Common/Git$
```

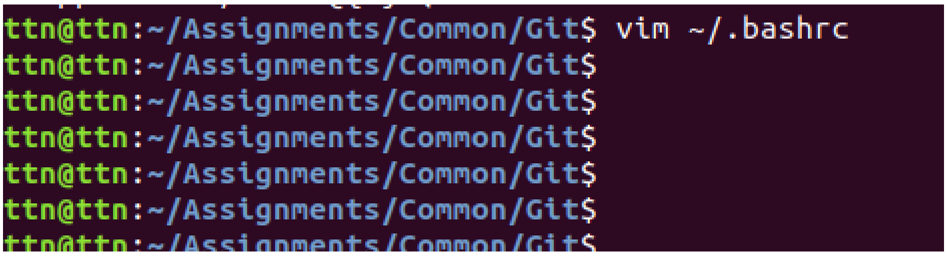
Q18. Add the following code to your .bashrc file :color_prompt="yes"

parse_git_branch() {

git branch 2> /dev/null | sed -e '/^[^*]/d' -e 's/* \(.*\)/(\1)/'

```
}
if [ "$color_prompt" = yes ]; then
PS1='\u@\h\[\033[00m\]:\[\033[01;34m\]\W\[\033[01;31m\] $(parse_git_branch)\[\
033[00m\]\$ '
else
PS1='\u@\h:\W $(parse_git_branch)\$ '
fi
unset color_prompt force_color_promp
```

Ans. Using command $vim ~/.bashrc, the file opens in the editor, then added the code as shown in the question.



```
ttn@ttn:~/Assignments/Common/Git$ vim ~/.bashrc
ttn@ttn:~/Assignments/Common/Git$
ttn@ttn:~/Assignments/Common/Git$
ttn@ttn:~/Assignments/Common/Git$
ttn@ttn:~/Assignments/Common/Git$
ttn@ttn:~/Assignments/Common/Git$
ttn@ttn:~/Assignments/Common/Git$
```