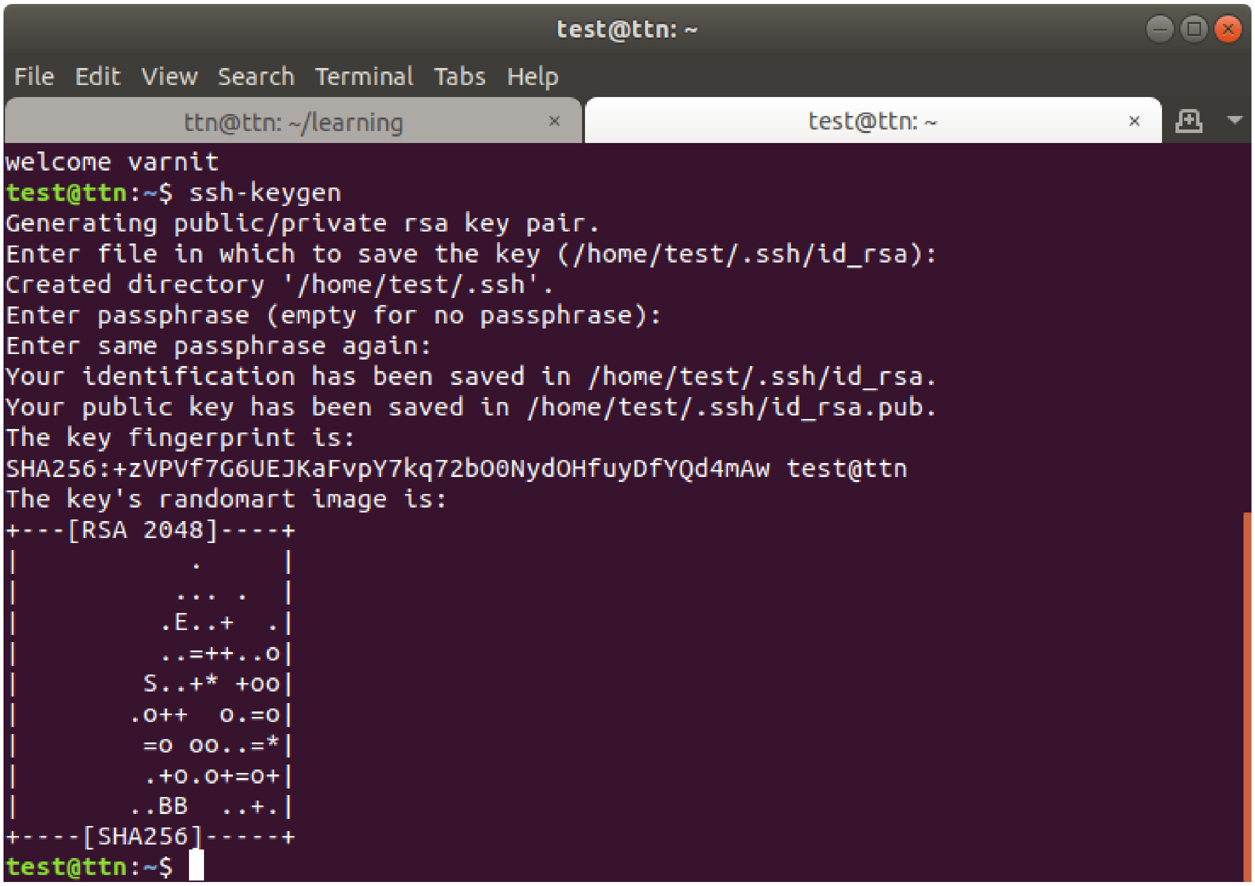
Version Control Exercise 5th Feb

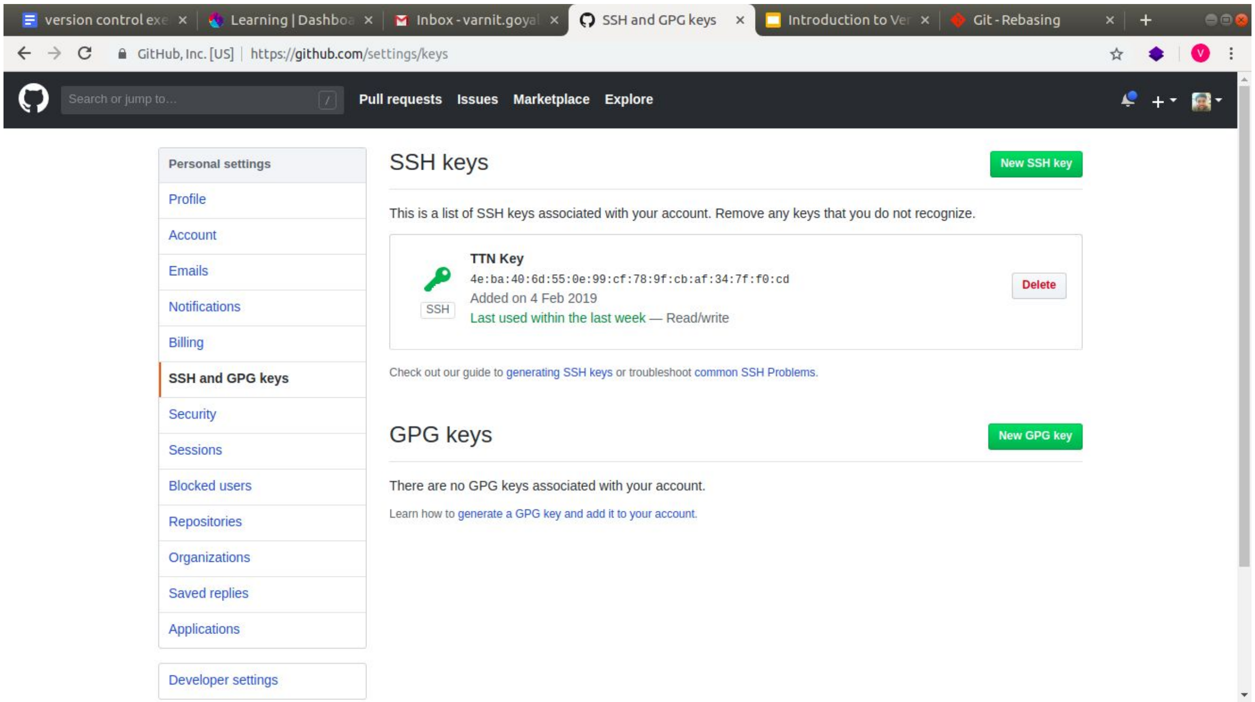
Q.1 setup a Git Setup https://confluence.atlassian.com/bitbucket/set-up-git-744723531.html
Ans-> I have first created an account on github and then generate my ssh keys on my ubuntu laptop using following commands

```
ssh-keygen
```

Then i copied the content of ~/.ssh/id_rsa.pub to github -> settings -> ssh -> add ssh key for passwordless authentication



```
welcome varnit
test@ttn:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/test/.ssh/id_rsa):
Created directory '/home/test/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/test/.ssh/id_rsa.
Your public key has been saved in /home/test/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:+zVPVf7G6UEJKaFvpY7kq72bO0NydOHfuyDfYQd4mAw test@ttn
The key's randomart image is:
+---[RSA 2048]----+
|           .     |
|          ... .  |
|         .E..+  .|
|         ..=++..o|
|        S..+* +oo|
|       .o++  o.=o|
|        =o oo..=*|
|        .+o.o+=o+|
|       ..BB  ..+.|
+----[SHA256]-----+
test@ttn:~$
```



SSH keys

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

TTN Key
4e:ba:40:6d:55:0e:99:cf:78:9f:cb:af:34:7f:f0:cd
Added on 4 Feb 2019
Last used within the last week — Read/write

Check out our guide to generating SSH keys or troubleshoot common SSH Problems.

GPG keys

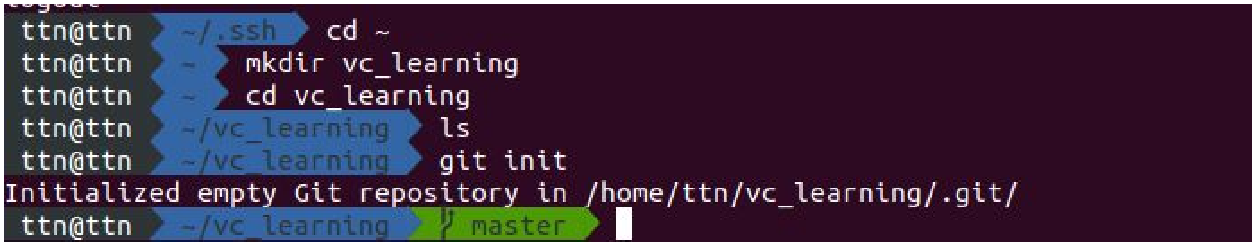
There are no GPG keys associated with your account.

Learn how to generate a GPG key and add it to your account.

Q.2 Initialize a Git Repository?

Ans->

```
cd ~/
mkdir vc_learning
cd vc_learning
git init
```



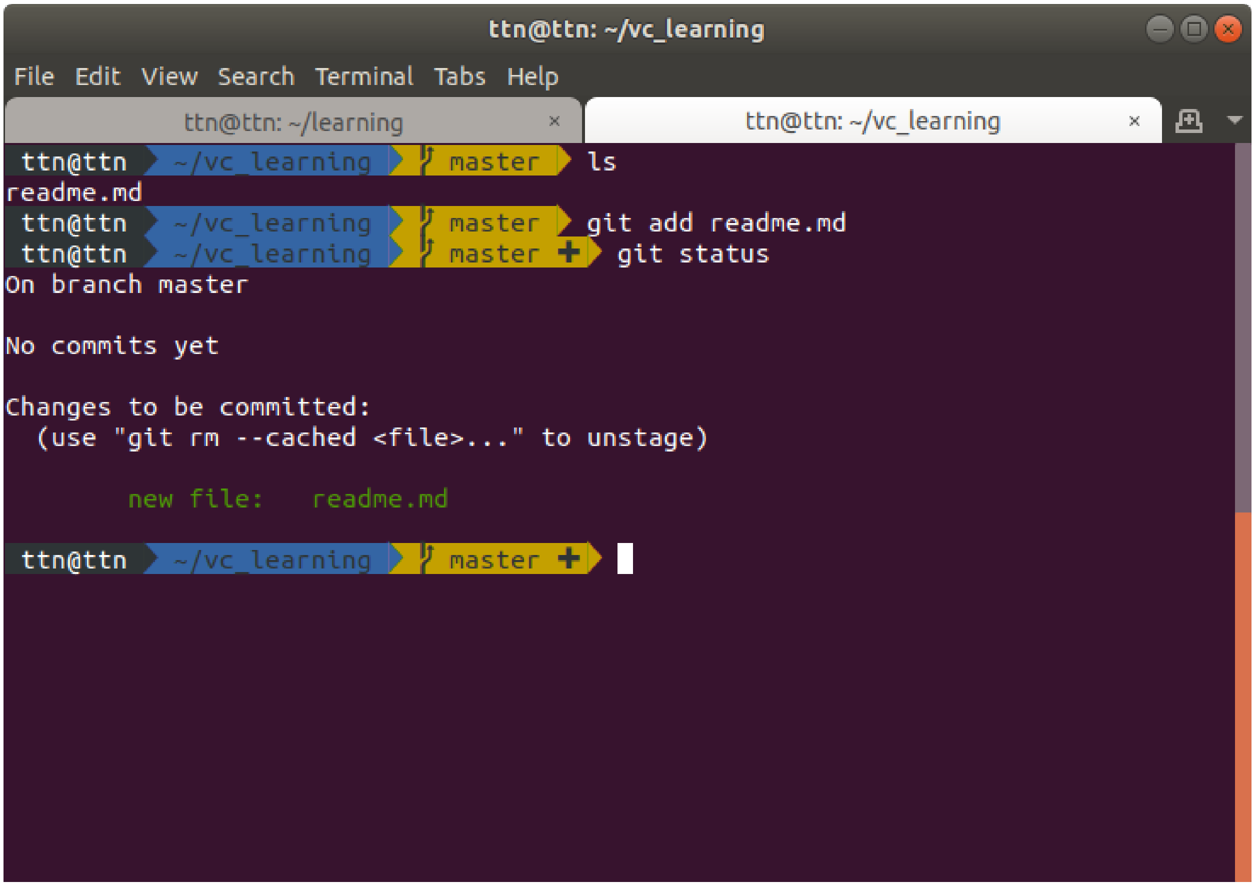
```
 ttn@ttn  ~/.ssh  cd ~
 ttn@ttn  ~  mkdir vc_learning
 ttn@ttn  ~  cd vc_learning
 ttn@ttn  ~/vc_learning  ls
 ttn@ttn  ~/vc_learning  git init
Initialized empty Git repository in /home/ttn/vc_learning/.git/
 ttn@ttn  ~/vc_learning  master 
```

Q.3 Add files to the repository

Ans-> i have created a readme.md file with some content inside it and then

```
git add readme.md
git status
```

Then used git status to check the status of the repository



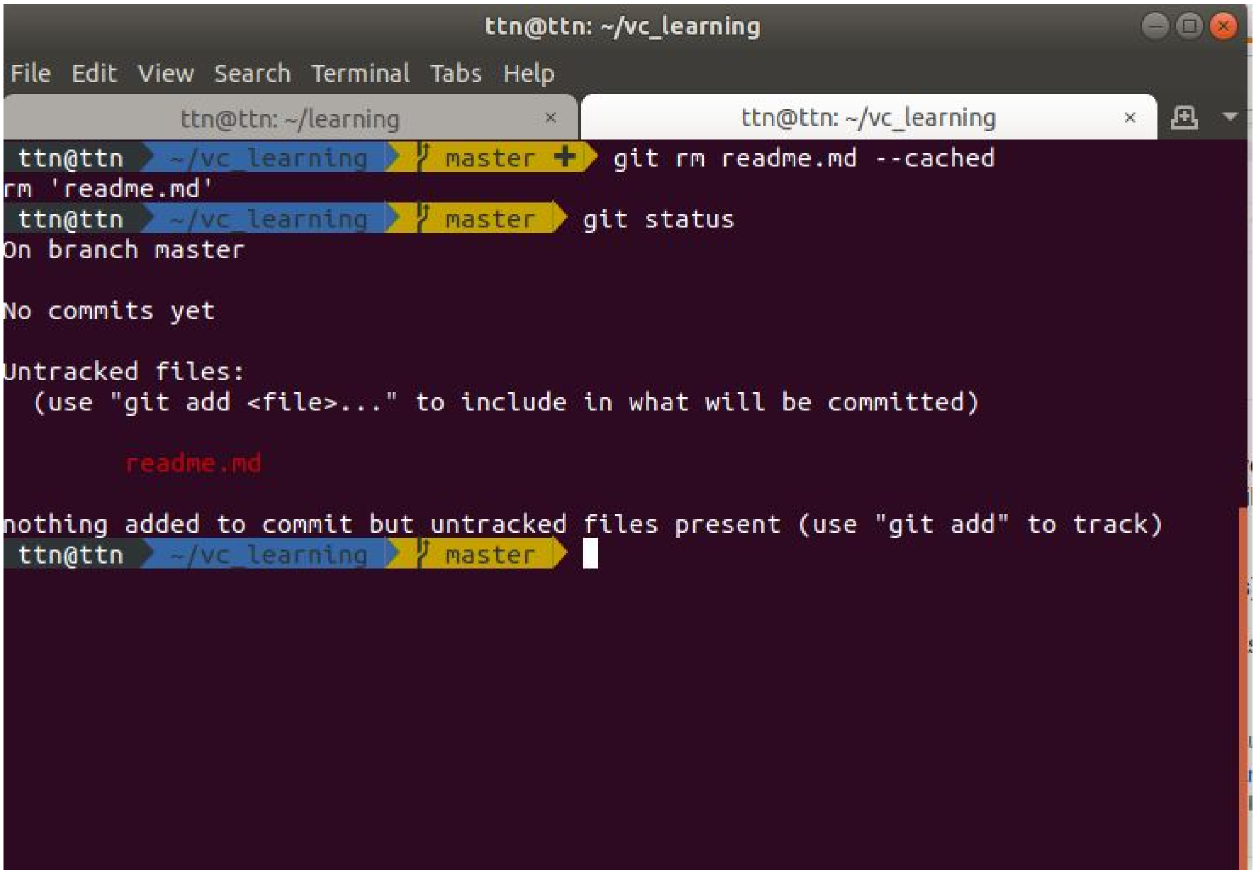
```
 ttn@ttn  ~/vc_learning  master  ls
readme.md
 ttn@ttn  ~/vc_learning  master  git add readme.md
 ttn@ttn  ~/vc_learning  master ✚  git status
On branch master

No commits yet

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

        new file:   readme.md

 ttn@ttn  ~/vc_learning  master ✚ 
```

Q.4 Unstage 1 file
Ans-> For unstaging a file i have used

```
git rm --cached readme.md
```

command then we checked the status of our repository with

```
git status
```



```
ttn@ttn  ~/vc_learning  master ✚  git rm readme.md --cached
rm 'readme.md'
ttn@ttn  ~/vc_learning  master  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)
ttn@ttn  ~/vc_learning  master 
```

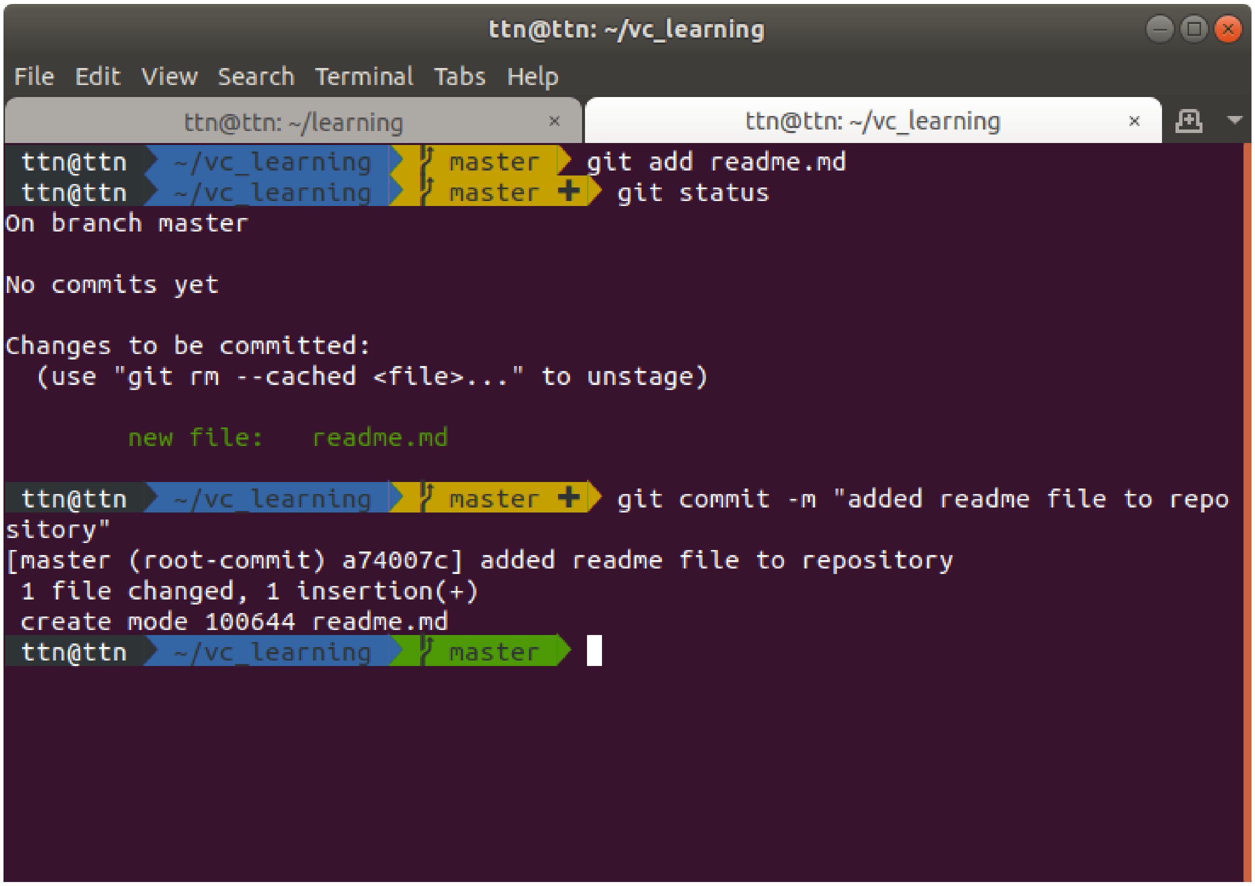
Q.5 Commit the file
Ans -> For committing the file i have first staged the file using

```
git add readme.md
```

Then

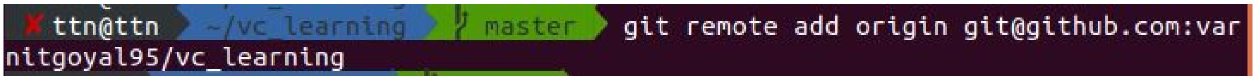
```
git commit -m "adding readme file to repository"
```

```
 git remote add origin git@github.com:varnitgoyal95/vc_learning.git
```



```
 ttn@ttn  ~/vc_learning   master  git add readme.md
 ttn@ttn  ~/vc_learning   master +  git status
On branch master

No commits yet

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

        new file:   readme.md

 ttn@ttn  ~/vc_learning   master +  git commit -m "added readme file to repo
sitory"
[master (root-commit) a74007c] added readme file to repository
 1 file changed, 1 insertion(+)
 create mode 100644 readme.md
 ttn@ttn  ~/vc_learning   master  
```

Q.6 Add a remote
Ans-> i have first created a new repository on github.com by the name vc_learning after than i executed the command on my local repository

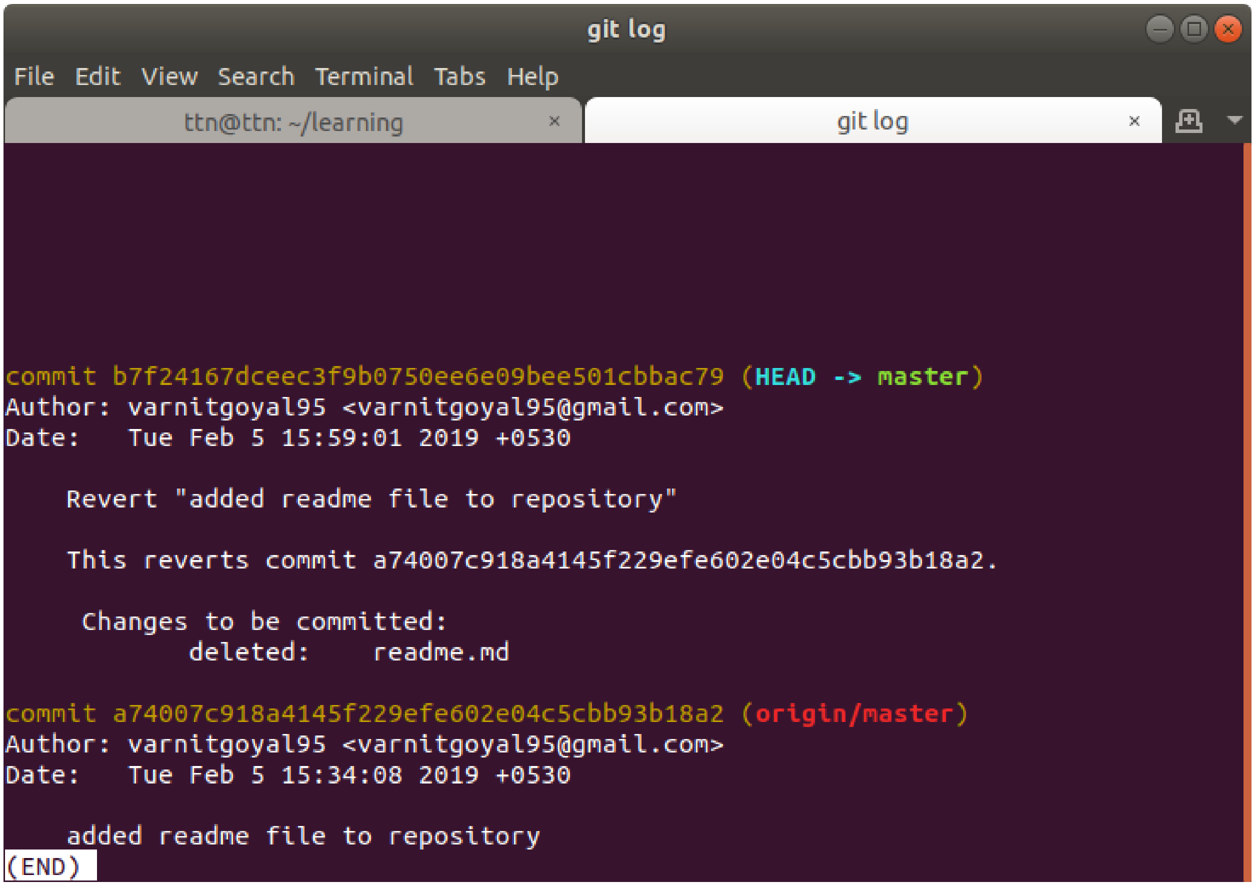


```
 ✘ ttn@ttn  ~/vc_learning   master  git remote add origin git@github.com:var
nitgoyal95/vc_learning
```

Q.7 Undo changes to a particular file

Ans -> To revert the changes we need to find commit hash of the commit we want to revert which can be found using

```
git log
```

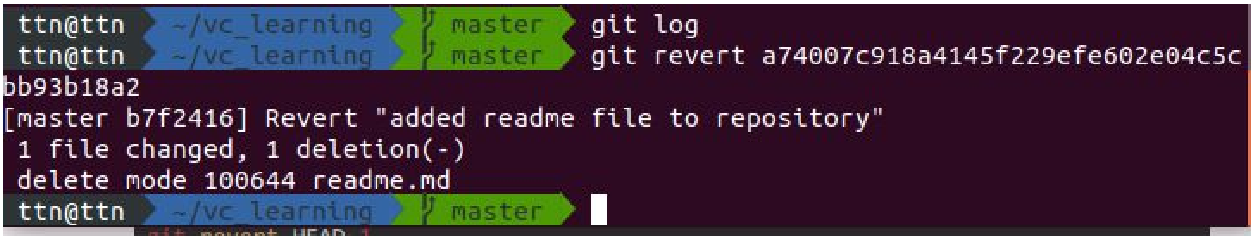


```
commit b7f24167dceec3f9b0750ee6e09bee501cbbac79 (HEAD -> master)
Author: varnitgoyal95 <varnitgoyal95@gmail.com>
Date:   Tue Feb 5 15:59:01 2019 +0530

    Revert "added readme file to repository"

    This reverts commit a74007c918a4145f229efe602e04c5cbb93b18a2.

     Changes to be committed:
            deleted:    readme.md

commit a74007c918a4145f229efe602e04c5cbb93b18a2 (origin/master)
Author: varnitgoyal95 <varnitgoyal95@gmail.com>
Date:   Tue Feb 5 15:34:08 2019 +0530

    added readme file to repository
(END)
```

So, we found the commit hash as a74007c918a4145f229efe602e04c5cbb93b18a2

Now to revert the changes i have executed the

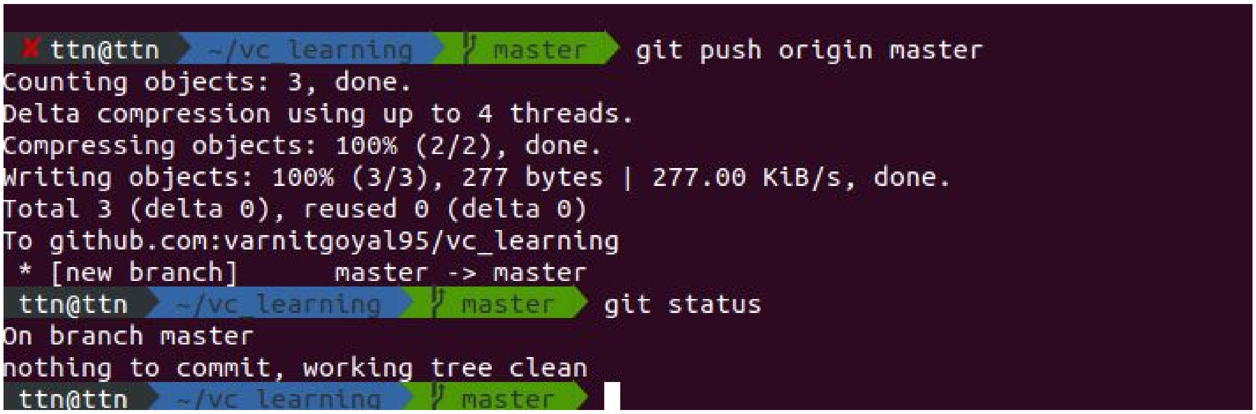
```
git revert a74007c918a4145f229efe602e04c5cbb93b18a2
```



```
 ttn@ttn  ~/vc_learning  master  git log
 ttn@ttn  ~/vc_learning  master  git revert a74007c918a4145f229efe602e04c5c
bb93b18a2
[master b7f2416] Revert "added readme file to repository"
 1 file changed, 1 deletion(-)
 delete mode 100644 readme.md
 ttn@ttn  ~/vc_learning  master 
```

Q.8 Push changes to Github.
Ans -> For pushing changes to github we use

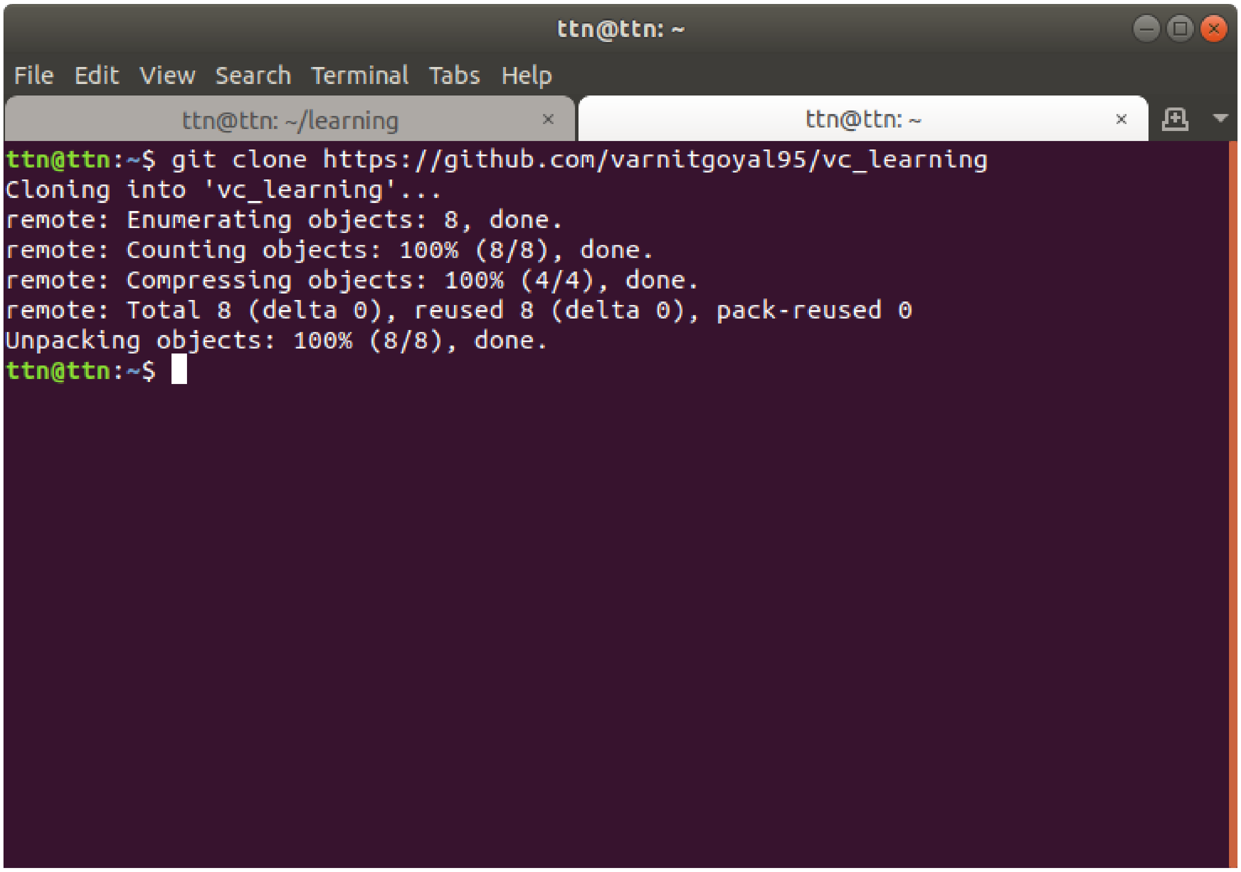
```
git push origin master
```



```
✘ ttn@ttn  ~/vc_learning  ᚴ master  git push origin master
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 277 bytes | 277.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To github.com:varnitgoyal95/vc_learning
 * [new branch]      master -> master
 ttn@ttn  ~/vc_learning  ᚴ master  git status
On branch master
nothing to commit, working tree clean
 ttn@ttn  ~/vc_learning  ᚴ master 
```

Q.9 Clone the repository
Ans-> For cloning existing repository we will use

```
git clone https://github.com/varnitgoyal95/vc_learning
```



```
ttn@ttn:~$ git clone https://github.com/varnitgoyal95/vc_learning
Cloning into 'vc_learning'...
remote: Enumerating objects: 8, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 8 (delta 0), reused 8 (delta 0), pack-reused 0
Unpacking objects: 100% (8/8), done.
ttn@ttn:~$
```

Q.10 Add changes to one of the copies and pull the changes in the other.
Ans-> we first made a folder using

```
mkdir vc_learning_copy
```

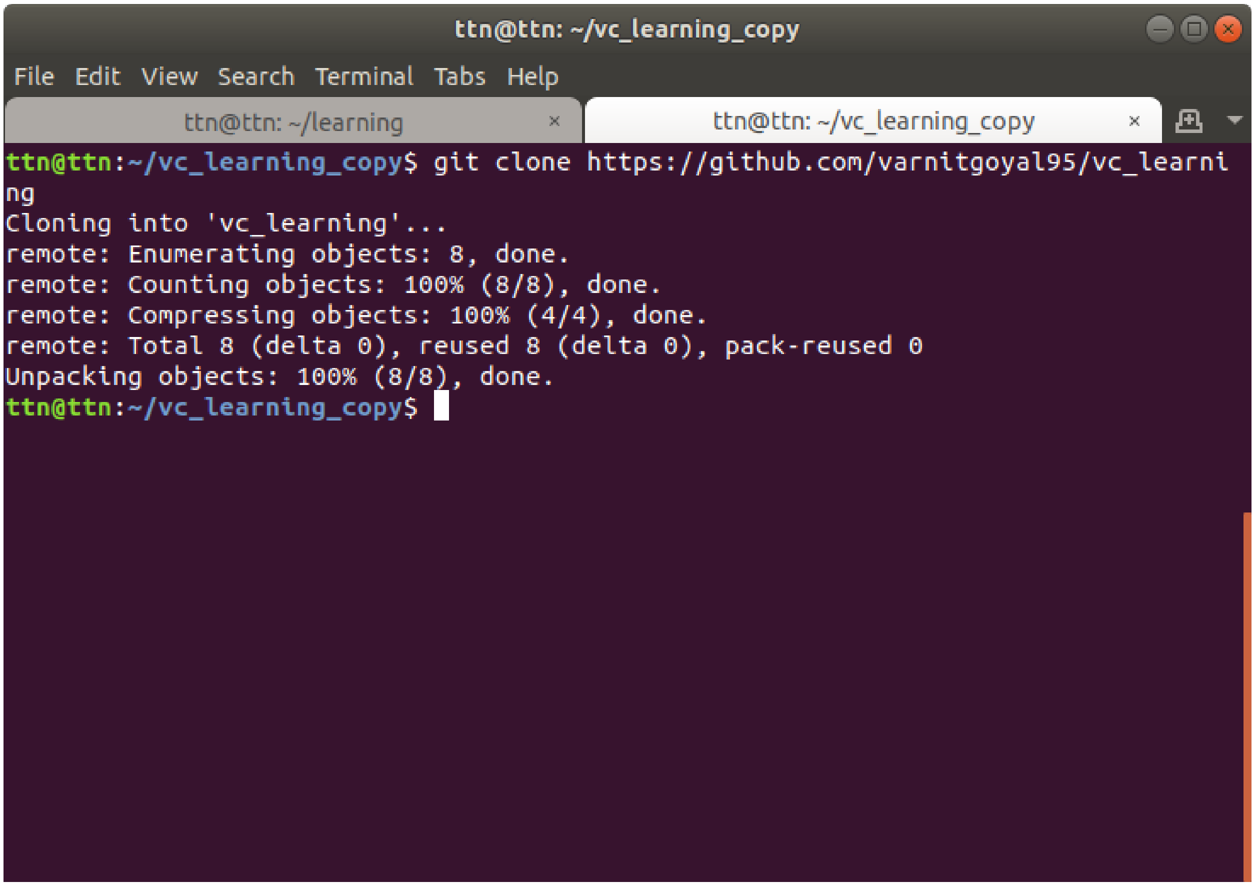
Then clone the repository in to that folder

```
git clone https://github.com/varnitgoyal95/vc_learning
```

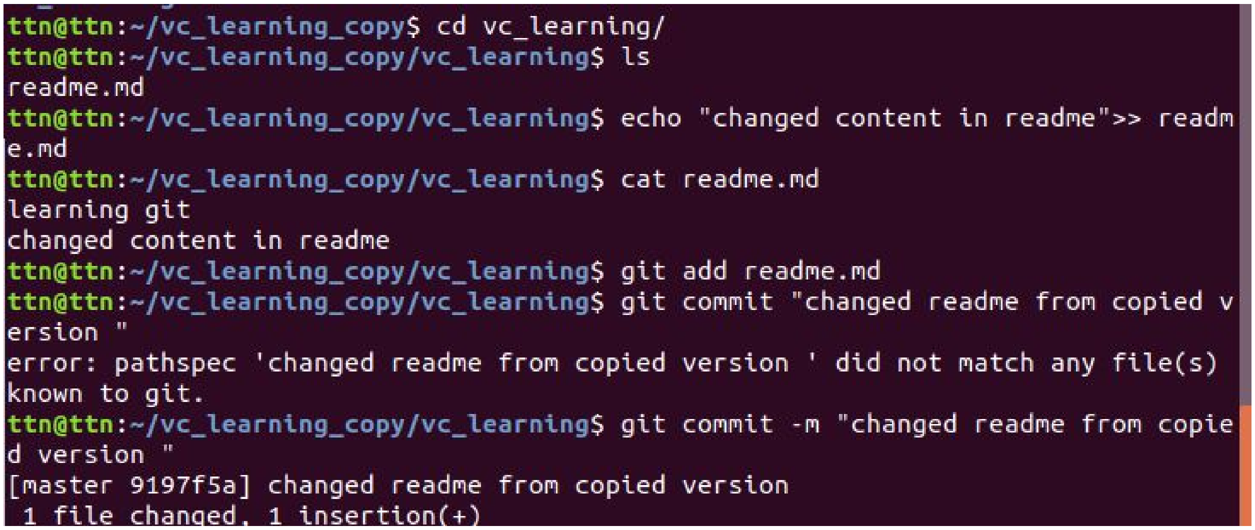
Then

```
cd  ~/vc_learning_copy/vc_learning
```

Then i made some changes in to the file then commit the changes and then push them to origin master


ttn@ttn: ~/vc_learning_copy
File Edit View Search Terminal Tabs Help
ttn@ttn: ~/learning
ttn@ttn: ~/vc_learning_copy

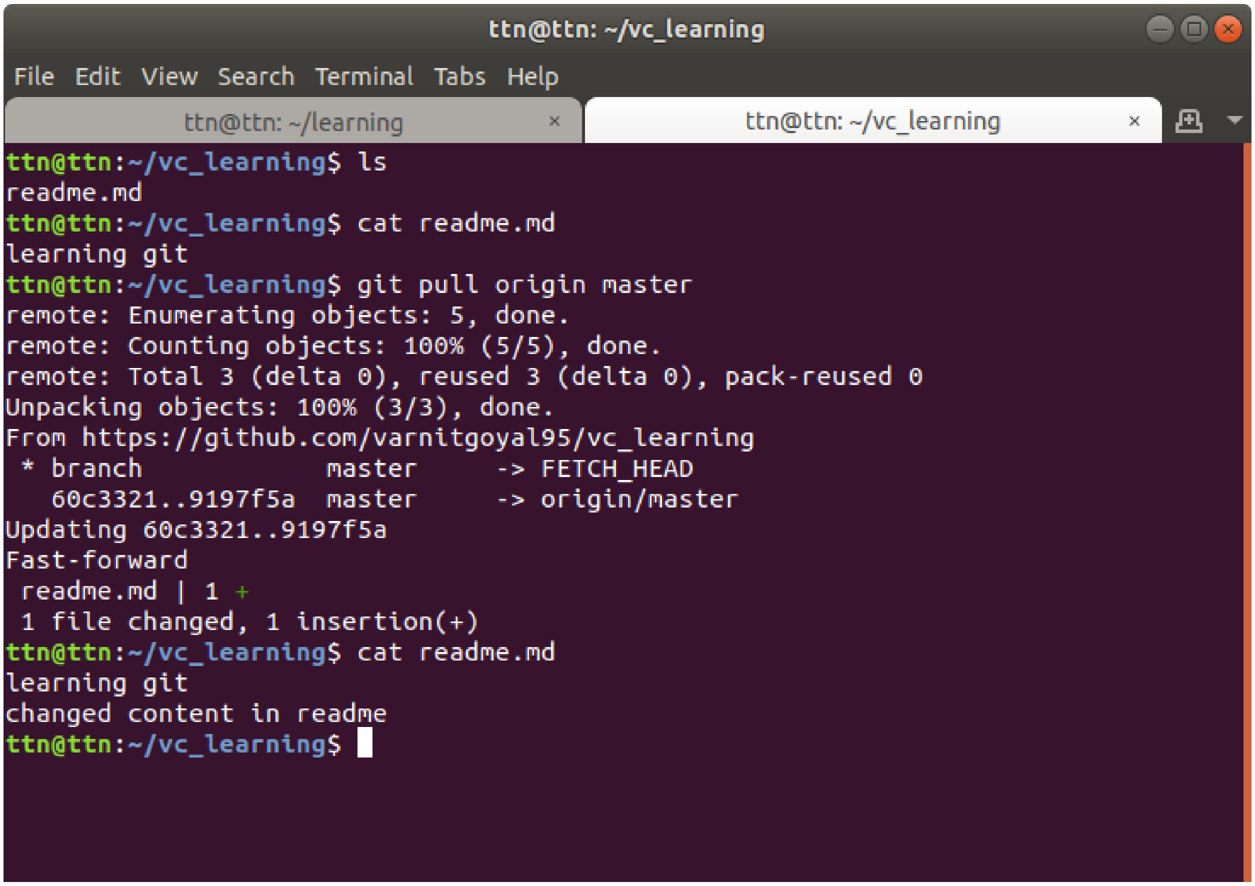
```
ttn@ttn:~/vc_learning_copy$ git clone https://github.com/varnitgoyal95/vc_learni
ng
Cloning into 'vc_learning'...
remote: Enumerating objects: 8, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 8 (delta 0), reused 8 (delta 0), pack-reused 0
Unpacking objects: 100% (8/8), done.
ttn@ttn:~/vc_learning_copy$ 
```



```
ttn@ttn:~/vc_learning_copy$ cd vc_learning/
ttn@ttn:~/vc_learning_copy/vc_learning$ ls
readme.md
ttn@ttn:~/vc_learning_copy/vc_learning$ echo "changed content in readme">> readm
e.md
ttn@ttn:~/vc_learning_copy/vc_learning$ cat readme.md
learning git
changed content in readme
ttn@ttn:~/vc_learning_copy/vc_learning$ git add readme.md
ttn@ttn:~/vc_learning_copy/vc_learning$ git commit "changed readme from copied v
ersion "
error: pathspec 'changed readme from copied version ' did not match any file(s)
known to git.
ttn@ttn:~/vc_learning_copy/vc_learning$ git commit -m "changed readme from copie
d version "
[master 9197f5a] changed readme from copied version
 1 file changed, 1 insertion(+)
```

Then, to pull the latest changes in the original folder

```
cd ~/vc_learning
git pull origin master
```

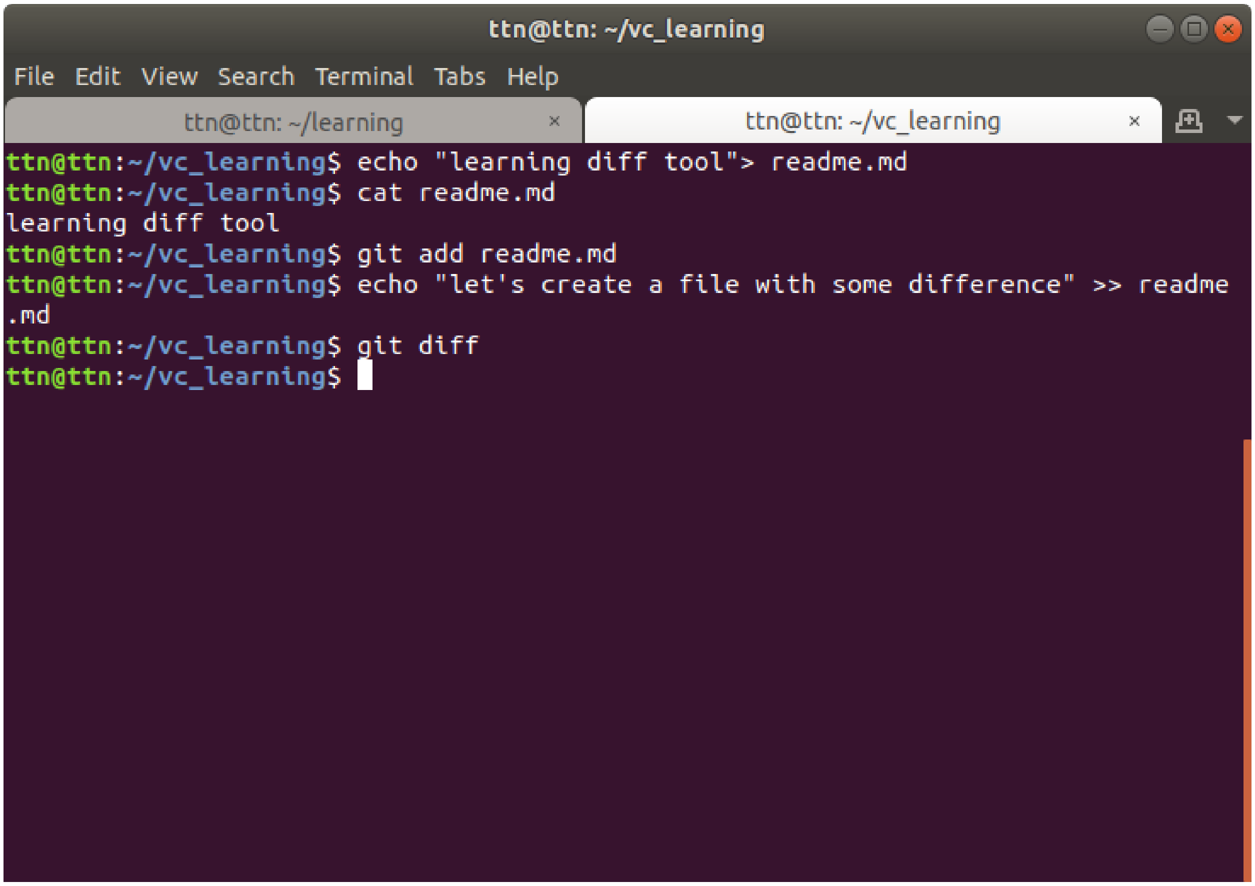


```
ttn@ttn:~/vc_learning$ ls
readme.md
ttn@ttn:~/vc_learning$ cat readme.md
learning git
ttn@ttn:~/vc_learning$ git pull origin master
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/varnitgoyal95/vc_learning
 * branch            master     -> FETCH_HEAD
   60c3321..9197f5a  master     -> origin/master
Updating 60c3321..9197f5a
Fast-forward
 readme.md | 1 +
 1 file changed, 1 insertion(+)
ttn@ttn:~/vc_learning$ cat readme.md
learning git
changed content in readme
ttn@ttn:~/vc_learning$ 
```

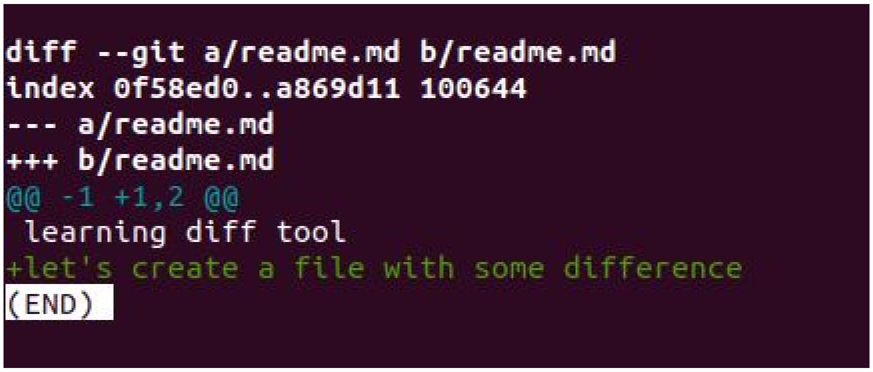
Q.11 Check differences between a file and its staged version

Ans-> first we changed our readme file and staged it using git add command then we changed it again and use

```
git diff --staged
```



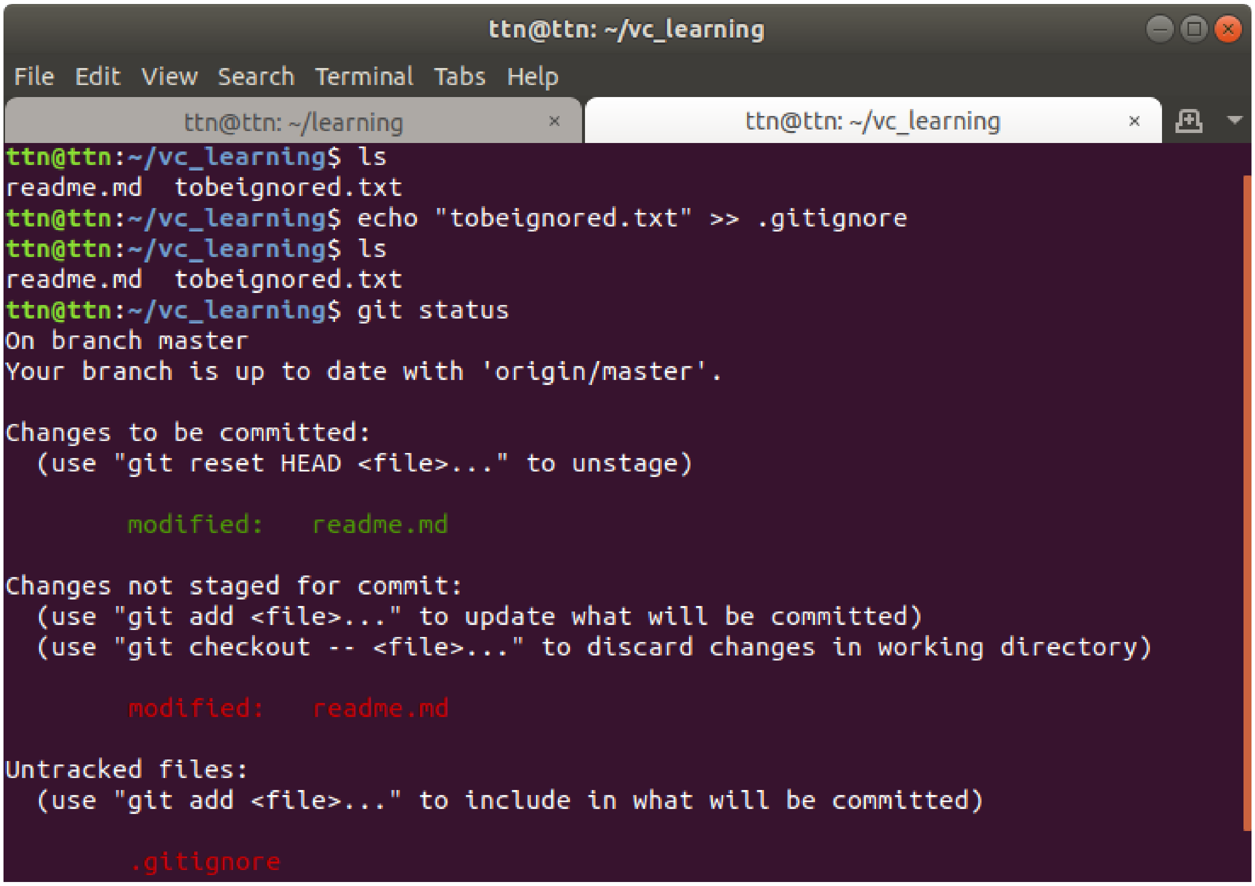
```
ttn@ttn:~/vc_learning$ echo "learning diff tool"> readme.md
ttn@ttn:~/vc_learning$ cat readme.md
learning diff tool
ttn@ttn:~/vc_learning$ git add readme.md
ttn@ttn:~/vc_learning$ echo "let's create a file with some difference" >> readme
.md
ttn@ttn:~/vc_learning$ git diff
ttn@ttn:~/vc_learning$ 
```



```
diff --git a/readme.md b/readme.md
index 0f58ed0..a869d11 100644
--- a/readme.md
+++ b/readme.md
@@ -1 +1,2 @@
 learning diff tool
+let's create a file with some difference
(END)
```

Q.12 Ignore a few files to be checked in

Ans -> To ignore some files we need a .gitignore file so, let’s create one named tobeignored.txt then create another file with the name of .gitignore and add tobeignored.txt into the list of ignored files

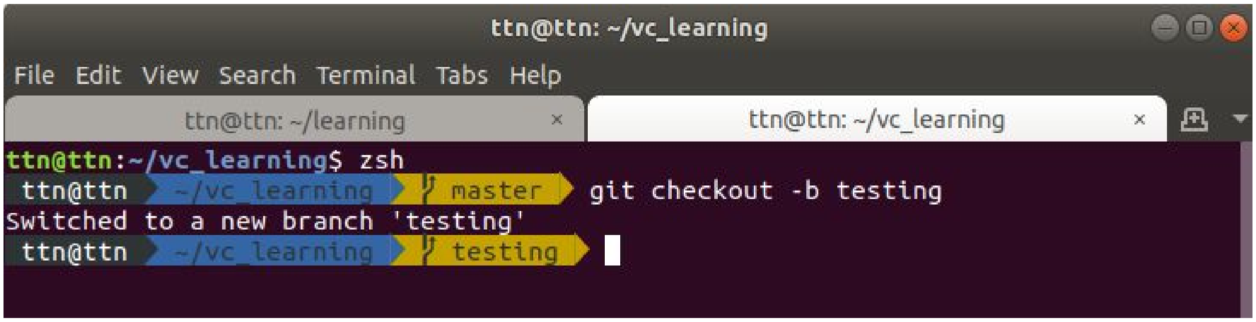


```
ttn@ttn:~/vc_learning$ ls
readme.md  tobeignored.txt
ttn@ttn:~/vc_learning$ echo "tobeignored.txt" >> .gitignore
ttn@ttn:~/vc_learning$ ls
readme.md  tobeignored.txt
ttn@ttn:~/vc_learning$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   readme.md

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:   readme.md

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

        .gitignore
```

Its evident from the screenshot that the files added into the .gitignore file is not visible in untracked files .

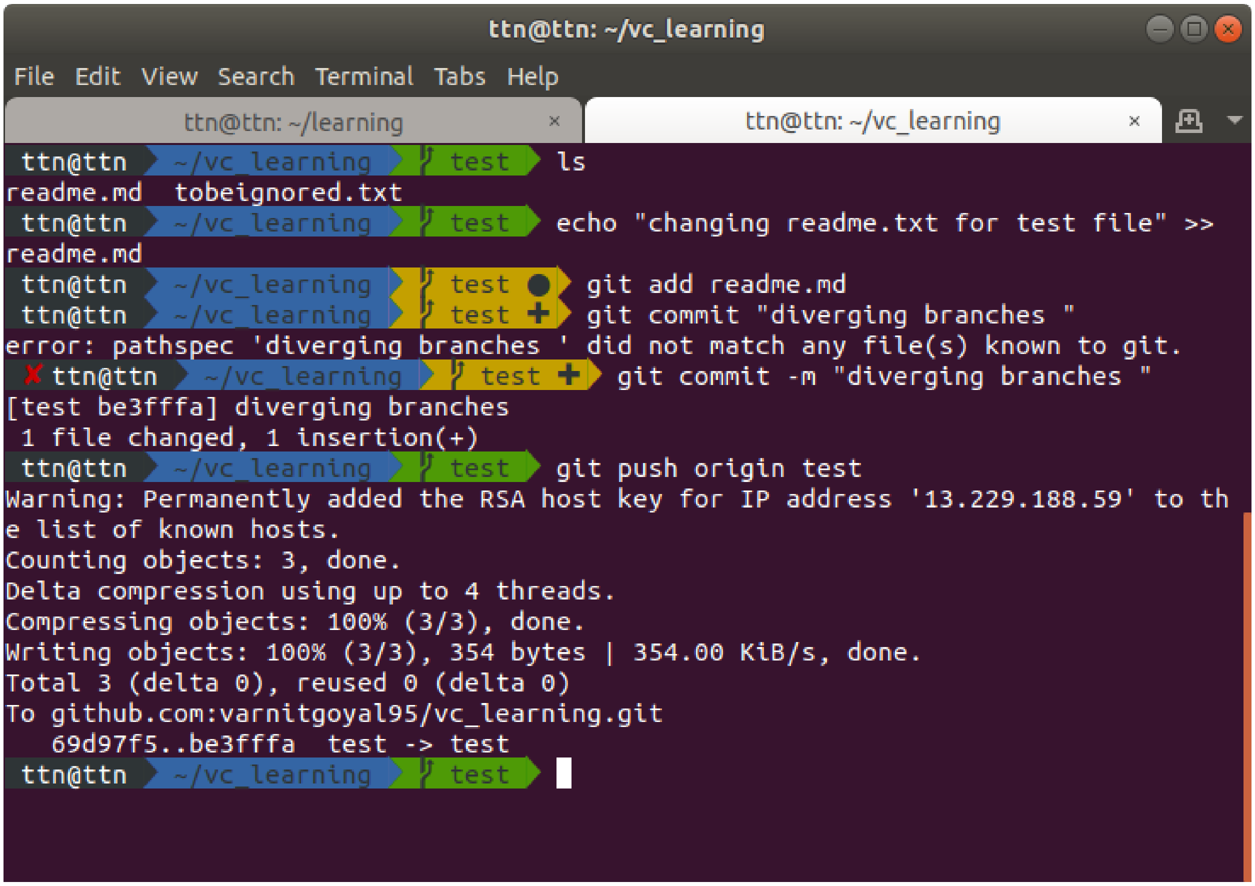
Q.13 Create a new branch.
Ans-> for creating a new branch and checkout to it we will use

```
git checkout -b test
```



```
ttn@ttn:~/vc_learning$ zsh
 ttn@ttn  ~/vc_learning  master  git checkout -b testing
Switched to a new branch 'testing'
 ttn@ttn  ~/vc_learning  testing 
```

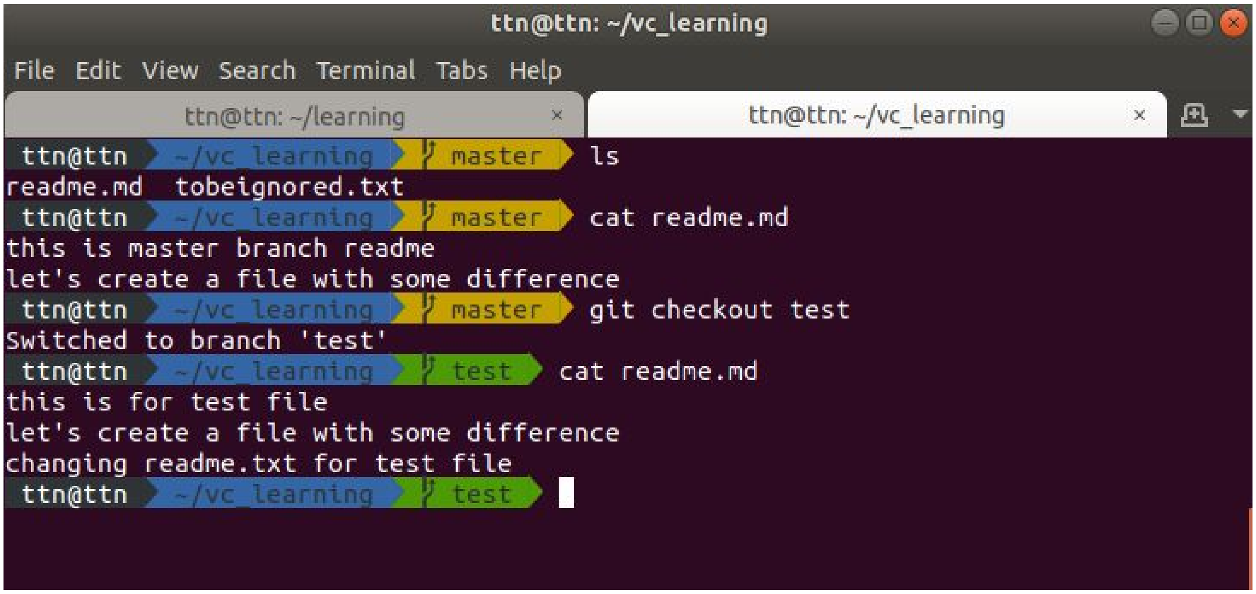
Q.14 Diverge them with commits
Ans-> Let’s diverge both of branches by making changes in the test branch



```
ttn@ttn  ~/vc_learning  test  ls
readme.md  tobeignored.txt
ttn@ttn  ~/vc_learning  test  echo "changing readme.txt for test file" >>
readme.md
ttn@ttn  ~/vc_learning  test ●  git add readme.md
ttn@ttn  ~/vc_learning  test ✚  git commit "diverging branches "
error: pathspec 'diverging branches ' did not match any file(s) known to git.
✘ ttn@ttn  ~/vc_learning  test ✚  git commit -m "diverging branches "
[test be3fffa] diverging branches
 1 file changed, 1 insertion(+)
ttn@ttn  ~/vc_learning  test  git push origin test
Warning: Permanently added the RSA host key for IP address '13.229.188.59' to th
e list of known hosts.
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 354 bytes | 354.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To github.com:varnitgoyal95/vc_learning.git
   69d97f5..be3fffa  test -> test
ttn@ttn  ~/vc_learning  test 
```

Q.15 Edit the same file at the same line on both branches and commit.

Ans-> edit the same file in test and master branch

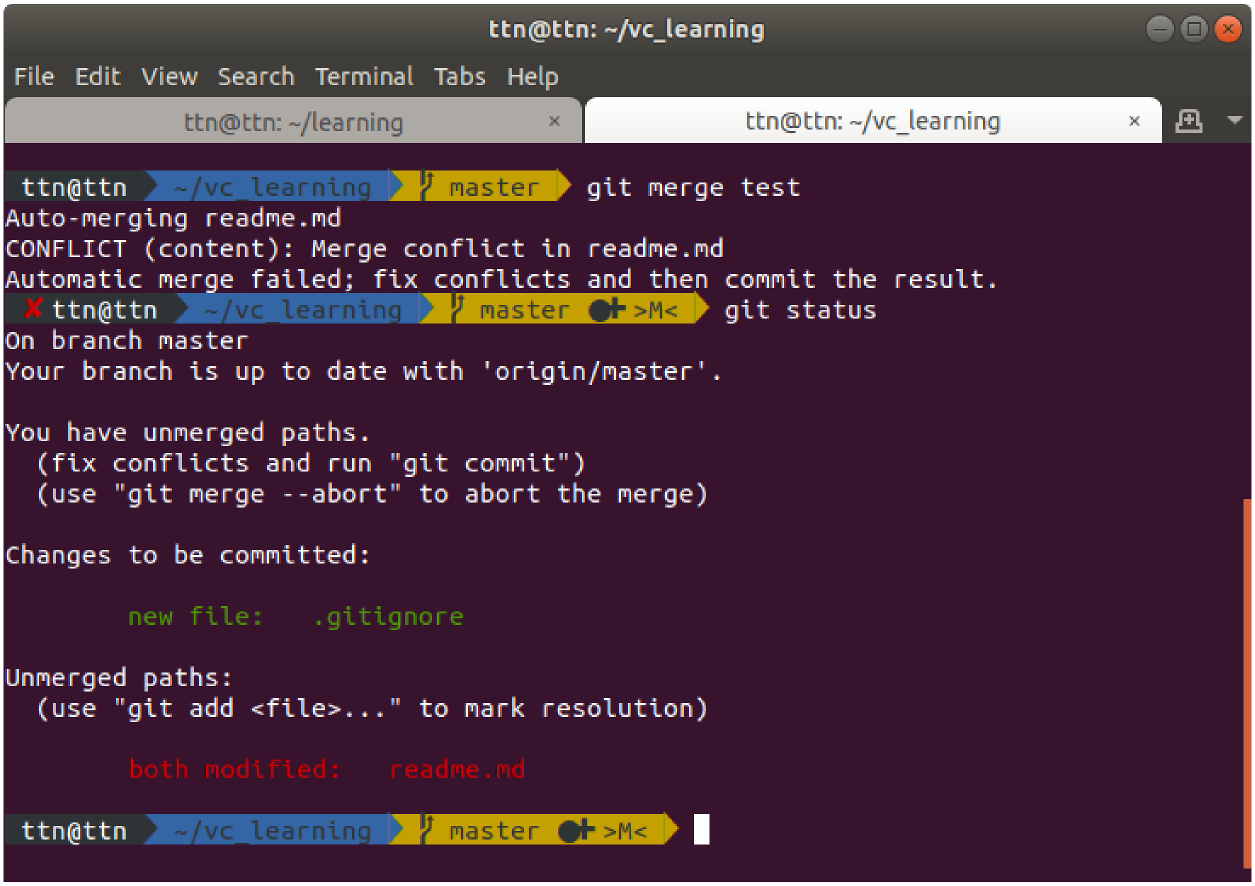


```
ttn@ttn  ~/vc_learning  master  ls
readme.md  tobeignored.txt
ttn@ttn  ~/vc_learning  master  cat readme.md
this is master branch readme
let's create a file with some difference
ttn@ttn  ~/vc_learning  master  git checkout test
Switched to branch 'test'
ttn@ttn  ~/vc_learning  test  cat readme.md
this is for test file
let's create a file with some difference
changing readme.txt for test file
ttn@ttn  ~/vc_learning  test 
```

Q.16 Try merging and resolve merge conflicts.

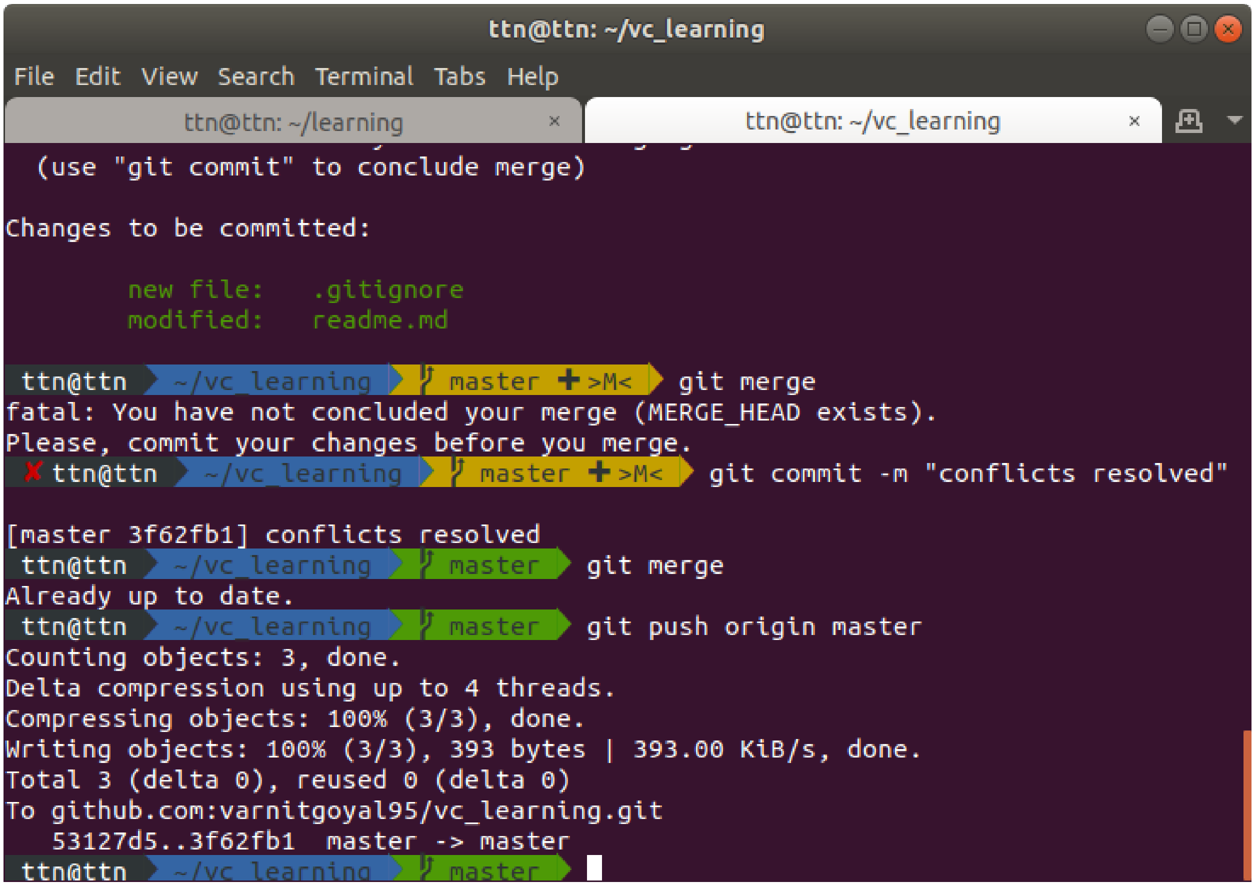
Ans-> we tried to merge the repo using

```
git merge test
```



```
 ttn@ttn  ~/vc_learning   master  git merge test
Auto-merging readme.md
CONFLICT (content): Merge conflict in readme.md
Automatic merge failed; fix conflicts and then commit the result.
 ✘ ttn@ttn  ~/vc_learning   master ●✚ >M<  git status
On branch master
Your branch is up to date with 'origin/master'.

You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Changes to be committed:

        new file:   .gitignore

Unmerged paths:
  (use "git add <file>..." to mark resolution)

        both modified:   readme.md

 ttn@ttn  ~/vc_learning   master ●✚ >M<  
```

Let’s resolve conflicts



```
  (use "git commit" to conclude merge)

Changes to be committed:

        new file:   .gitignore
        modified:   readme.md

 ttn@ttn  ~/vc_learning  master ✚ >M<  git merge
fatal: You have not concluded your merge (MERGE_HEAD exists).
Please, commit your changes before you merge.
 ✘ ttn@ttn  ~/vc_learning  master ✚ >M<  git commit -m "conflicts resolved"

[master 3f62fb1] conflicts resolved
 ttn@ttn  ~/vc_learning  master  git merge
Already up to date.
 ttn@ttn  ~/vc_learning  master  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), 393 bytes | 393.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To github.com:varnitgoyal95/vc_learning.git
   53127d5..3f62fb1  master -> master
 ttn@ttn  ~/vc_learning  master 
```

Q.17 Stash the changes and pop them

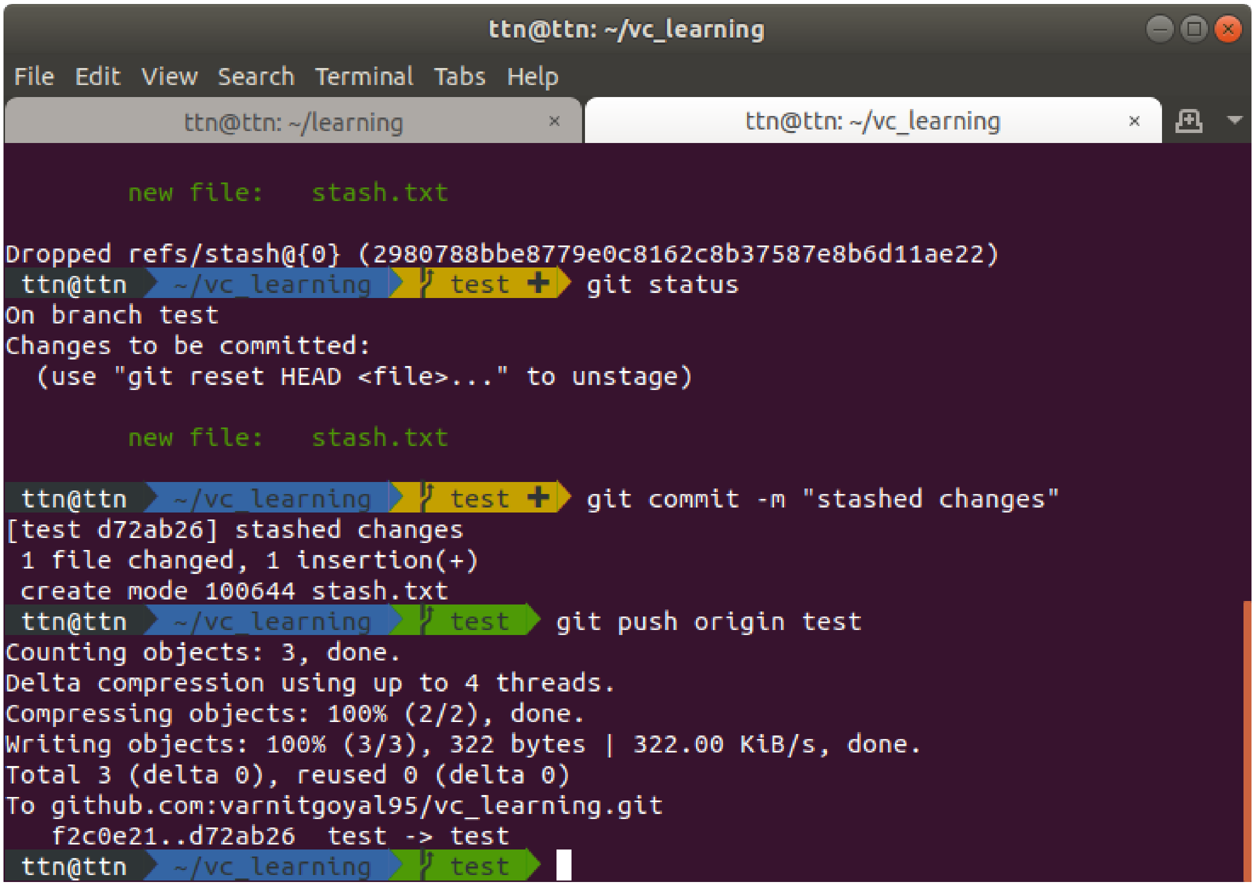
Ans -> we added a new file called stash.txt and added the file to master branch then we stashed the changes using

```
git stash
git checkout test
```

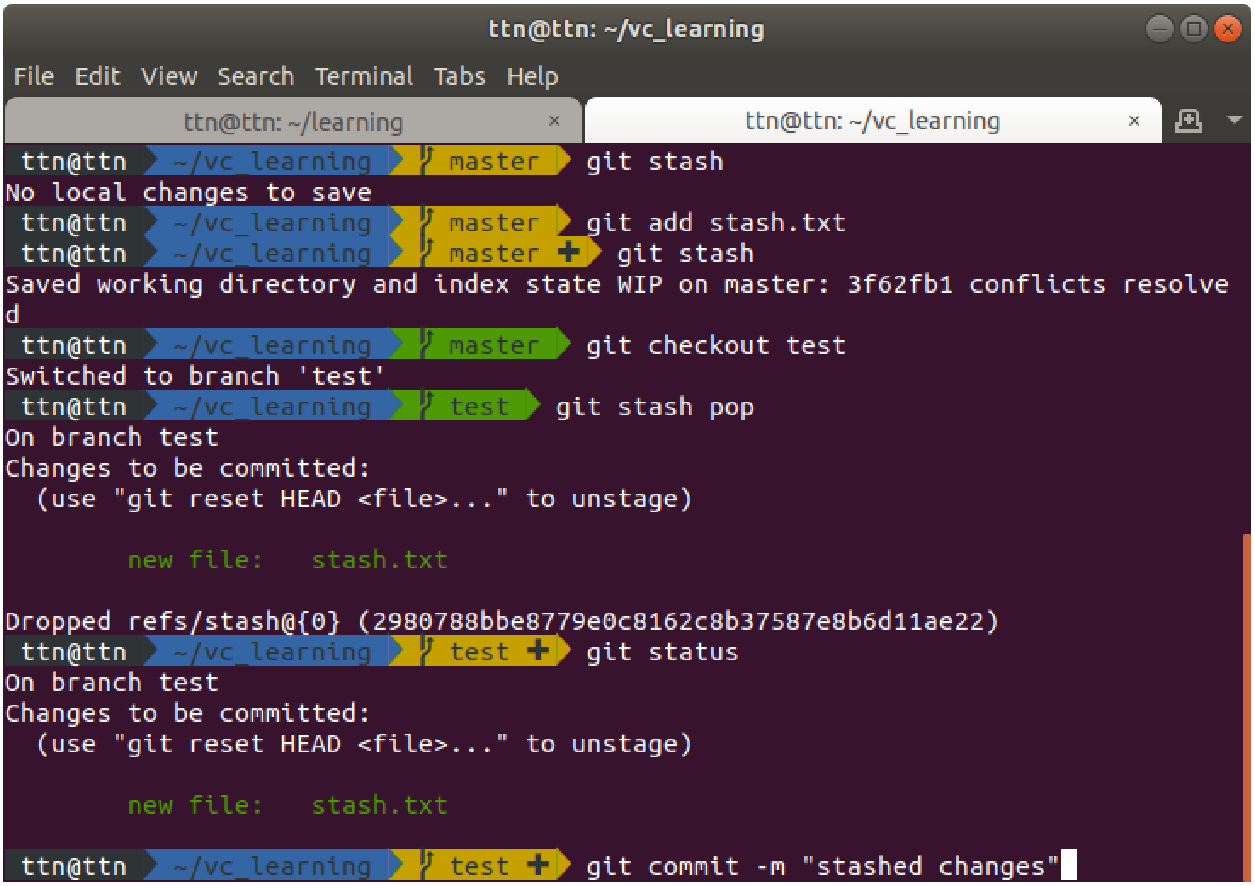
Then pop the changes in test branch with

```
git stash pop
```

Then commit them to new test branch



```
        new file:   stash.txt

Dropped refs/stash@{0} (2980788bbe8779e0c8162c8b37587e8b6d11ae22)
 ttn@ttn  ~/vc_learning   test ✚  git status
On branch test
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   stash.txt

 ttn@ttn  ~/vc_learning   test ✚  git commit -m "stashed changes"
[test d72ab26] stashed changes
 1 file changed, 1 insertion(+)
 create mode 100644 stash.txt
 ttn@ttn  ~/vc_learning   test  git push origin test
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 322 bytes | 322.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To github.com:varnitgoyal95/vc_learning.git
   f2c0e21..d72ab26  test -> test
 ttn@ttn  ~/vc_learning   test 
```

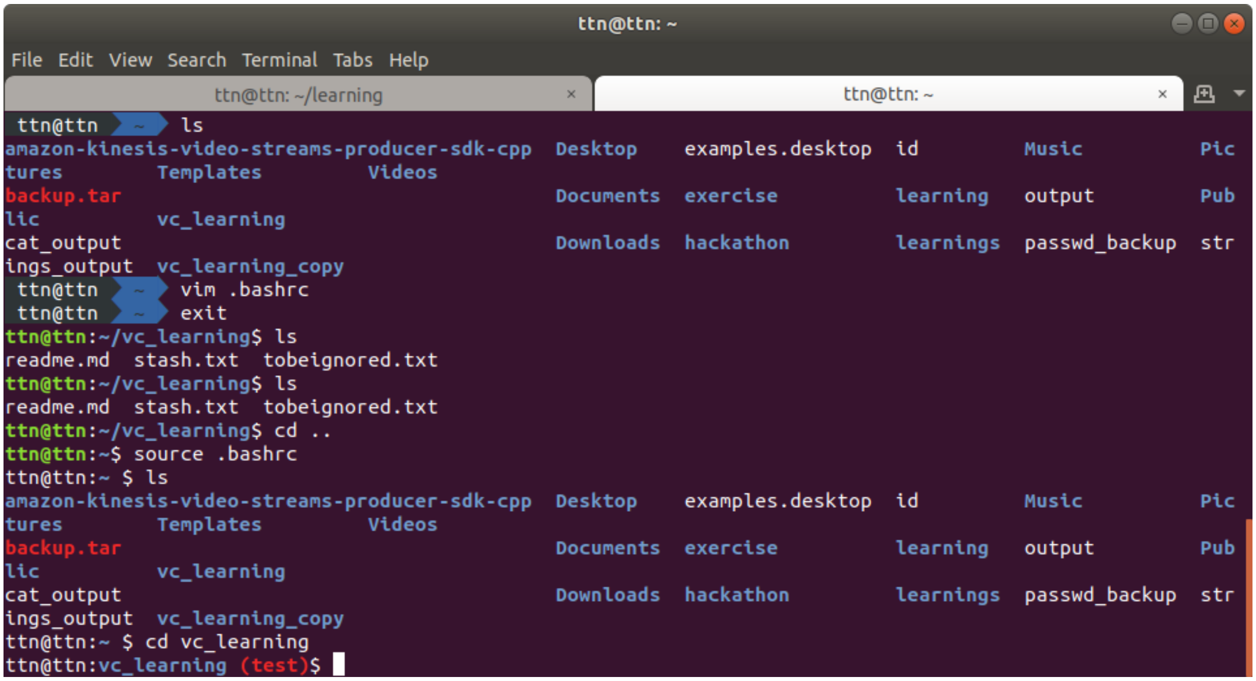


```
ttn@ttn  ~/vc_learning  master  git stash
No local changes to save
ttn@ttn  ~/vc_learning  master  git add stash.txt
ttn@ttn  ~/vc_learning  master ✚  git stash
Saved working directory and index state WIP on master: 3f62fb1 conflicts resolve
d
ttn@ttn  ~/vc_learning  master  git checkout test
Switched to branch 'test'
ttn@ttn  ~/vc_learning  test  git stash pop
On branch test
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   stash.txt

Dropped refs/stash@{0} (2980788bbe8779e0c8162c8b37587e8b6d11ae22)
ttn@ttn  ~/vc_learning  test ✚  git status
On branch test
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   stash.txt

ttn@ttn  ~/vc_learning  test ✚  git commit -m "stashed changes"
```

Q.18 Add the following code to your .bashrc file : color_prompt="yes"

Ans-> let’s add the following code into 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_prompt
```



```
 ttn@ttn  ~  ls
amazon-kinesis-video-streams-producer-sdk-cpp  Desktop    examples.desktop  id         Music          Pic
tures         Templates          Videos
backup.tar                                     Documents  exercise          learning   output         Pub
lic           vc_learning
cat_output                                     Downloads  hackathon         learnings  passwd_backup  str
ings_output  vc_learning_copy
 ttn@ttn  ~  vim .bashrc
 ttn@ttn  ~  exit
ttn@ttn:~/vc_learning$ ls
readme.md  stash.txt  tobeignored.txt
ttn@ttn:~/vc_learning$ ls
readme.md  stash.txt  tobeignored.txt
ttn@ttn:~/vc_learning$ cd ..
ttn@ttn:~$ source .bashrc
ttn@ttn:~ $ ls
amazon-kinesis-video-streams-producer-sdk-cpp  Desktop    examples.desktop  id         Music          Pic
tures         Templates          Videos
backup.tar                                     Documents  exercise          learning   output         Pub
lic           vc_learning
cat_output                                     Downloads  hackathon         learnings  passwd_backup  str
ings_output  vc_learning_copy
ttn@ttn:~ $ cd vc_learning
ttn@ttn:vc_learning (test)$ 
```

After adding bashrc code we are able to see which branch we are working on in brackets