

1. Git Setup
(Already done)

```
ttn@ttn:~$ git version
git version 2.17.1
ttn@ttn:~$
```

2. Initialise Git repository

```
ttn@ttn:~$ mkdir Desktop/Assignment2
ttn@ttn:~$ cd Desktop/Assignment2/
ttn@ttn:~/Desktop/Assignment2$ git init
Initialized empty Git repository in /home/ttn/Desktop/Assignment2/.git/
ttn@ttn:~/Desktop/Assignment2$
```

3. Add files to repository

```
ttn@ttn:~/Desktop/Assignment2$ touch file1
ttn@ttn:~/Desktop/Assignment2$ ll
total 12
drwxr-xr-x 3 ttn ttn 4096 Feb  5 21:34 ./
drwxr-xr-x 5 ttn ttn 4096 Feb  5 21:34 ../
-rw-r--r-- 1 ttn ttn    0 Feb  5 21:34 file1
drwxr-xr-x 7 ttn ttn 4096 Feb  5 21:33 .git/
ttn@ttn:~/Desktop/Assignment2$ touch file2
ttn@ttn:~/Desktop/Assignment2$ git status
On branch master

No commits yet

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

        file1
        file2

nothing added to commit but untracked files present (use "git add" to track)
ttn@ttn:~/Desktop/Assignment2$ git add .
```

4. Unstage one file

```
ttn@ttn:~/Desktop/Assignment2$ git rm --cached file1
rm 'file1'
ttn@ttn:~/Desktop/Assignment2$
```

5. Commit the file

git commit -m "Added file2"

6. Add a remote

```
ttn@ttn:~/Desktop/Assignment2$ git remote add origin git@github.com:anuranjanbose/Assignment2.git
ttn@ttn:~/Desktop/Assignment2$ git remote -v
origin  git@github.com:anuranjanbose/Assignment2.git (fetch)
origin  git@github.com:anuranjanbose/Assignment2.git (push)
```

7. Undo changes to a particular file

```
ttn@ttn:~/Desktop/Assignment2$ nano file2
ttn@ttn:~/Desktop/Assignment2$ git checkout file2
ttn@ttn:~/Desktop/Assignment2$ cat file2
ttn@ttn:~/Desktop/Assignment2$
```

8. Push changes to GitHub

```
ttn@ttn:~/Desktop/Assignment2$ touch file3
ttn@ttn:~/Desktop/Assignment2$ git add file3
ttn@ttn:~/Desktop/Assignment2$ git commit -m "Added file3"
[master c08a2f7] Added file3
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 file3
ttn@ttn:~/Desktop/Assignment2$ git push origin master
Warning: Permanently added the RSA host key for IP address '64:ff9b::c01e:fd71'
to the list of known hosts.
Counting objects: 5, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (5/5), 417 bytes | 417.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0)
To github.com:anuranjanbose/Assignment2.git
 * [new branch]      master -> master
```

9. Clone the repository

```
ttn@ttn:~/Desktop/Assignment2$ cd ..
ttn@ttn:~/Desktop$ git clone git@github.com:anuranjanbose/Assignment2.git
fatal: destination path 'Assignment2' already exists and is not an empty directory.
ttn@ttn:~/Desktop$ mkdir VC2
ttn@ttn:~/Desktop$ cd VC2/
ttn@ttn:~/Desktop/VC2$ git clone git@github.com:anuranjanbose/Assignment2.git
Cloning into 'Assignment2'...
Warning: Permanently added the RSA host key for IP address '64:ff9b::c01e:fd70'
to the list of known hosts.
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 5 (delta 0), reused 5 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), done.
```

10. Add changes to one copy and pull from another

```
ttn@ttn:~/Desktop/VC2$ cd Assignment2/
ttn@ttn:~/Desktop/VC2/Assignment2$ touch file4
ttn@ttn:~/Desktop/VC2/Assignment2$ git add file4
ttn@ttn:~/Desktop/VC2/Assignment2$ git commit -m "Added file4"
[master 4c6bac4] Added file4
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 file4
ttn@ttn:~/Desktop/VC2/Assignment2$ git push origin master
Counting objects: 2, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 242 bytes | 242.00 KiB/s, done.
Total 2 (delta 0), reused 0 (delta 0)
To github.com:anuranjanbose/Assignment2.git
c08a2f7..4c6bac4 master -> master
ttn@ttn:~/Desktop/VC2/Assignment2$ cd ..
ttn@ttn:~/Desktop/VC2$ cd ..
ttn@ttn:~/Desktop$ cd Assignment2/
ttn@ttn:~/Desktop/Assignment2$ git pull origin master
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 2 (delta 0), reused 2 (delta 0), pack-reused 0
Unpacking objects: 100% (2/2), done.
remote: Total 2 (delta 0), reused 2 (delta 0), pack-reused 0
Unpacking objects: 100% (2/2), done.
From github.com:anuranjanbose/Assignment2
* branch          master      -> FETCH_HEAD
c08a2f7..4c6bac4 master      -> origin/master
Updating c08a2f7..4c6bac4
Fast-forward
 file4 | 0
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 file4
```

11. Check differences between a file and staged version

```
ttn@ttn:~/Desktop$ cd VC2/
ttn@ttn:~/Desktop/VC2$ cd Assignment2/
ttn@ttn:~/Desktop/VC2/Assignment2$ touch file1
ttn@ttn:~/Desktop/VC2/Assignment2$ echo "hi" > file1
ttn@ttn:~/Desktop/VC2/Assignment2$ git add file1
ttn@ttn:~/Desktop/VC2/Assignment2$ git diff --cached
diff --git a/file1 b/file1
new file mode 100644
index 00000000..45b983b
--- /dev/null
+++ b/file1
@@ -0,0 +1 @@
+hi
ttn@ttn:~/Desktop/VC2/Assignment2$
```

12. Ignore few files to be checked in

```
ttn@ttn:~/Desktop/VC2/Assignment2$ nano .gitignore
ttn@ttn:~/Desktop/VC2/Assignment2$ cat .gitignore
file5
ttn@ttn:~/Desktop/VC2/Assignment2$ touch file5
ttn@ttn:~/Desktop/VC2/Assignment2$ 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)

    new file:   file1

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

    .gitignore
```

13. Create a new branch

```
ttn@ttn:~/Desktop/VC2/Assignment2$ git checkout -b branch1
A      file1
Switched to a new branch 'branch1'
ttn@ttn:~/Desktop/VC2/Assignment2$ git branch
* branch1
  master
ttn@ttn:~/Desktop/VC2/Assignment2$
```

14. Diverge them with commits

```
ttn@ttn:~/Desktop/VC2/Assignment2$ echo "hi" > temp_file
ttn@ttn:~/Desktop/VC2/Assignment2$ git add temp_file
ttn@ttn:~/Desktop/VC2/Assignment2$ git commit -m "temp_file at branch1"
[branch1 ddee902] temp_file at branch1
 2 files changed, 2 insertions(+)
 create mode 100644 file1
 create mode 100644 temp_file
ttn@ttn:~/Desktop/VC2/Assignment2$ git checkout branch2
error: pathspec 'branch2' did not match any file(s) known to git.
ttn@ttn:~/Desktop/VC2/Assignment2$ git checkout -b branch2
Switched to a new branch 'branch2'
ttn@ttn:~/Desktop/VC2/Assignment2$ echo "hi" > temp_file
ttn@ttn:~/Desktop/VC2/Assignment2$ git add temp_file
ttn@ttn:~/Desktop/VC2/Assignment2$ git commit -m "temp_file at branch2"
On branch branch2
Untracked files:
  .gitignore

nothing added to commit but untracked files present
ttn@ttn:~/Desktop/VC2/Assignment2$ git add .
ttn@ttn:~/Desktop/VC2/Assignment2$ git commit -m "temp_file at branch2"
[branch2 c17f385] temp_file at branch2
 1 file changed, 1 insertion(+)
 create mode 100644 .gitignore
```


15. Edit the same file on both branches and commit

```
ttn@ttn:~/Desktop/VC2/Assignment2$ git checkout branch1
Switched to branch 'branch1'
ttn@ttn:~/Desktop/VC2/Assignment2$ echo "hi branch1" > file
ttn@ttn:~/Desktop/VC2/Assignment2$ git add file
ttn@ttn:~/Desktop/VC2/Assignment2$ git commit
Aborting commit due to empty commit message.
ttn@ttn:~/Desktop/VC2/Assignment2$ git commit -m "file at branch1"
[branch1 7e4b1b8] file at branch1
1 file changed, 1 insertion(+), 1 deletion(-)
ttn@ttn:~/Desktop/VC2/Assignment2$ git checkout branch2
Switched to branch 'branch2'
ttn@ttn:~/Desktop/VC2/Assignment2$ echo "hi branch2" > file
ttn@ttn:~/Desktop/VC2/Assignment2$ git add file
ttn@ttn:~/Desktop/VC2/Assignment2$ git commit -m "file at branch2"
[branch2 ad1b65c] file at branch2
1 file changed, 1 insertion(+), 1 deletion(-)
ttn@ttn:~/Desktop/VC2/Assignment2$
```

16. Try merging and resolve conflicts

```
ttn@ttn:~/Desktop/VC2/Assignment2$ git merge branch1
Auto-merging file
CONFLICT (add/add): Merge conflict in file
Automatic merge failed; fix conflicts and then commit the result.
ttn@ttn:~/Desktop/VC2/Assignment2$ cat file
<<<<<<< HEAD
hi branch2
=====
hi branch1
>>>>>>> branch1
ttn@ttn:~/Desktop/VC2/Assignment2$ nano file
ttn@ttn:~/Desktop/VC2/Assignment2$ cat file
hi branch2
hi branch1
ttn@ttn:~/Desktop/VC2/Assignment2$ git add .
ttn@ttn:~/Desktop/VC2/Assignment2$ git commit -m "conflict resolved"
[branch2 17b8651] conflict resolved
ttn@ttn:~/Desktop/VC2/Assignment2$
```

17. Stash the changes and pop them

```
ttn@ttn:~/Desktop/VC2/Assignment2$ git status
On branch branch2
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:   file

no changes added to commit (use "git add" and/or "git commit -a")
ttn@ttn:~/Desktop/VC2/Assignment2$ git add .
ttn@ttn:~/Desktop/VC2/Assignment2$ git stash
Saved working directory and index state WIP on branch2: 17b8651 conflict resolved
ttn@ttn:~/Desktop/VC2/Assignment2$ git stash list
stash@{0}: WIP on branch2: 17b8651 conflict resolved
ttn@ttn:~/Desktop/VC2/Assignment2$ git stash pop
On branch branch2
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:   file

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

18. 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_prompt
```

```
ttn@ttn:~$ sudo nano .bashrc
[sudo] password for ttn:
```