

Ans 1

```
shashank@TTNPL-shashankpandey:~/test$  
shashank@TTNPL-shashankpandey:~/test$  
shashank@TTNPL-shashankpandey:~/test$ git --version  
git version 2.34.1  
shashank@TTNPL-shashankpandey:~/test$
```

Ans 2

```
shashank@TTNPL-shashankpandey:~/test$ cd test-repo  
shashank@TTNPL-shashankpandey:~/test/test-repo$ git init  
hint: Using 'master' as the name for the initial branch. This default branch name  
hint: is subject to change. To configure the initial branch name to use in all  
hint: of your new repositories, which will suppress this warning, call:  
hint:  
hint:   git config --global init.defaultBranch <name>  
hint:  
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and  
hint: 'development'. The just-created branch can be renamed via this command:  
hint:  
hint:   git branch -m <name>  
Initialized empty Git repository in /home/shashank/test/test-repo/.git/  
shashank@TTNPL-shashankpandey:~/test/test-repo$
```

Ans 3

```
hint:   git branch -m <name>  
Initialized empty Git repository in /home/shashank/test/test-repo/.git/  
shashank@TTNPL-shashankpandey:~/test/test-repo$ touch sample.txt  
shashank@TTNPL-shashankpandey:~/test/test-repo$ echo "hello world" > sample.txt  
shashank@TTNPL-shashankpandey:~/test/test-repo$ git add .  
shashank@TTNPL-shashankpandey:~/test/test-repo$
```

Ans 4

```
nothing to commit, working tree clean
shashank@TTNPL-shashankpandey:~/test/test-repo$ echo "helllllloooo" >> sample.txt
shashank@TTNPL-shashankpandey:~/test/test-repo$ 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:   sample.txt

no changes added to commit (use "git add" and/or "git commit -a")
shashank@TTNPL-shashankpandey:~/test/test-repo$ git add .
shashank@TTNPL-shashankpandey:~/test/test-repo$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   sample.txt

shashank@TTNPL-shashankpandey:~/test/test-repo$ git restore --staged sample.txt
shashank@TTNPL-shashankpandey:~/test/test-repo$ 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:   sample.txt

no changes added to commit (use "git add" and/or "git commit -a")
shashank@TTNPL-shashankpandey:~/test/test-repo$
```

Ans 5

```
no changes added to commit (use "git add" and/or "git commit -a")
shashank@TTNPL-shashankpandey:~/test/test-repo$ git add sample.txt
shashank@TTNPL-shashankpandey:~/test/test-repo$ git commit -m "first comment : added sample.txt file"
[master fe6bbd2] first comment : added sample.txt file
1 file changed, 1 insertion(+)
shashank@TTNPL-shashankpandey:~/test/test-repo$
```

Ans 6

```
shashank@TTNPL-shashankpandey:~/bootcamp-test-repo$ git remote remove origin
shashank@TTNPL-shashankpandey:~/bootcamp-test-repo$ git remote -v
shashank@TTNPL-shashankpandey:~/bootcamp-test-repo$ git remote add origin https://github.com/shashankp10-TTN/bootcamp.git
```

Ans 7

Undo changes to a particular file [does this mean going back to the previous commit?]

```

shashank@TTNPL-shashankpandey:~/test/demo$
shashank@TTNPL-shashankpandey:~/test/demo$ git log
commit a5079e6f5c0c3ecda121b6d848684027992b7fde (HEAD)
Author: shashankp10-TTN <shashank.pandey@tothenew.com>
Date: Thu Jan 23 16:46:34 2025 +0530

    updated main.txt file

commit 852851291edbddf87cf4f6b0c85a80905f72f0fd
Author: shashankp10-TTN <shashank.pandey@tothenew.com>
Date: Thu Jan 23 16:41:54 2025 +0530

    added
shashank@TTNPL-shashankpandey:~/test/demo$ git reset 852851291edbddf87cf4f6b0c85a80905f72f0fd
Unstaged changes after reset:
M   main.txt
shashank@TTNPL-shashankpandey:~/test/demo$

```

Ans 8

```

shashank@TTNPL-shashankpandey:~/test$ git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 347 bytes | 347.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:shashankp10-TTN/bootcamp.git
   3f6544d..9dac624  main -> main
shashank@TTNPL-shashankpandey:~/test$

```

Ans 9

```

shashank@TTNPL-shashankpandey:~/test$ git clone git@github.com:shashankp10-TTN/bootcamp.git
Cloning into 'bootcamp'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 6 (delta 0), reused 6 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (6/6), done.
shashank@TTNPL-shashankpandey:~/test$

```

Ans 10

```
shashank@TTNPL-shashankpandey:~/test$ touch tst.txt
shashank@TTNPL-shashankpandey:~/test$ echo "hi" > tst.txt
shashank@TTNPL-shashankpandey:~/test$ git add .
shashank@TTNPL-shashankpandey:~/test$ git commit -m "added"
[main d869629] added
 1 file changed, 1 insertion(+)
 create mode 100644 tst.txt
shashank@TTNPL-shashankpandey:~/test$ git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 275 bytes | 275.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:shashankp10-TTN/bootcamp.git
   85a87bd..d869629  main -> main
shashank@TTNPL-shashankpandey:~/test$ git checkout p1
Switched to branch 'p1'
shashank@TTNPL-shashankpandey:~/test$ git pull origin main
From github.com:shashankp10-TTN/bootcamp
 * branch            main            -> FETCH_HEAD
Updating 85a87bd..d869629
Fast-forward
 tst.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 tst.txt
shashank@TTNPL-shashankpandey:~/test$
```

Ans 11

```
shashank@TTNPL-shashankpandey:~/test/demo$ touch new.txt
shashank@TTNPL-shashankpandey:~/test/demo$ git add new.txt
shashank@TTNPL-shashankpandey:~/test/demo$ echo "hiii" > new.txt
shashank@TTNPL-shashankpandey:~/test/demo$ git diff --staged
diff --git a/new.txt b/new.txt
new file mode 100644
index 0000000..e69de29
shashank@TTNPL-shashankpandey:~/test/demo$
```

Ans 12

```
shashank@TTNPL-shashankpandey:~/test/demo$  
shashank@TTNPL-shashankpandey:~/test/demo$ touch a.txt  
shashank@TTNPL-shashankpandey:~/test/demo$ touch b.txt  
shashank@TTNPL-shashankpandey:~/test/demo$ touch c.txt  
shashank@TTNPL-shashankpandey:~/test/demo$ git status  
On branch master  
Untracked files:  
  (use "git add <file>..." to include in what will be committed)  
    a.txt  
    b.txt  
    c.txt  
  
nothing added to commit but untracked files present (use "git add" to track)  
shashank@TTNPL-shashankpandey:~/test/demo$ echo "a.txt" > .gitignore  
shashank@TTNPL-shashankpandey:~/test/demo$ git status  
On branch master  
Untracked files:  
  (use "git add <file>..." to include in what will be committed)  
    .gitignore  
    b.txt  
    c.txt  
  
nothing added to commit but untracked files present (use "git add" to track)  
shashank@TTNPL-shashankpandey:~/test/demo$
```

Ans 13

```
shashank@TTNPL-shashankpandey:~/test$ git checkout -b p1  
Switched to a new branch 'p1'  
shashank@TTNPL-shashankpandey:~/test$ git pull origin main
```

Ans 14

```
shashank@TTNPL-shashankpandey:~/test/demo$ git checkout -b test-brach
Switched to a new branch 'test-brach'
shashank@TTNPL-shashankpandey:~/test/demo$ touch testFile.txt
shashank@TTNPL-shashankpandey:~/test/demo$ git add .
shashank@TTNPL-shashankpandey:~/test/demo$ git commit -m "added new file testFile.txt"
On branch test-brach
nothing to commit, working tree clean
shashank@TTNPL-shashankpandey:~/test/demo$ echo "changed" > testFile.txt
shashank@TTNPL-shashankpandey:~/test/demo$ git add .
shashank@TTNPL-shashankpandey:~/test/demo$ git commit -m "added content"
[test-brach 832f6e6] added content
1 file changed, 1 insertion(+)
shashank@TTNPL-shashankpandey:~/test/demo$ echo "changed1" > testFile.txt
shashank@TTNPL-shashankpandey:~/test/demo$ git add .
shashank@TTNPL-shashankpandey:~/test/demo$ git commit -m "added content again"
[test-brach f463f1a] added content again
1 file changed, 1 insertion(+), 1 deletion(-)
shashank@TTNPL-shashankpandey:~/test/demo$
```

Ans 15,16

```
shashank@TTNPL-shashankpandey:~/test$ mkdir demo
shashank@TTNPL-shashankpandey:~/test$ cd demo
shashank@TTNPL-shashankpandey:~/test/demo$ git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /home/shashank/test/demo/.git/
shashank@TTNPL-shashankpandey:~/test/demo$ touch main.txt
shashank@TTNPL-shashankpandey:~/test/demo$ git add .
shashank@TTNPL-shashankpandey:~/test/demo$ git commit -m "added"
[master (root-commit) 8528512] added
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 main.txt
shashank@TTNPL-shashankpandey:~/test/demo$ git checkout -b conflict
Switched to a new branch 'conflict'
shashank@TTNPL-shashankpandey:~/test/demo$ echo "heloo" > main.txt
shashank@TTNPL-shashankpandey:~/test/demo$ git add .
shashank@TTNPL-shashankpandey:~/test/demo$ git commit -m "conflict file added"
[conflict 50afc0d] conflict file added
1 file changed, 1 insertion(+)
shashank@TTNPL-shashankpandey:~/test/demo$ git checkout main
error: pathspec 'main' did not match any file(s) known to git
shashank@TTNPL-shashankpandey:~/test/demo$ git checkout master
Switched to branch 'master'
shashank@TTNPL-shashankpandey:~/test/demo$ git status
On branch master
nothing to commit, working tree clean
shashank@TTNPL-shashankpandey:~/test/demo$ echo 'hiii' > main.txt
shashank@TTNPL-shashankpandey:~/test/demo$ git add .
shashank@TTNPL-shashankpandey:~/test/demo$ git commit -m "updated main.txt file"
[master a5079e6] updated main.txt file
1 file changed, 1 insertion(+)
shashank@TTNPL-shashankpandey:~/test/demo$ git checkout conflict
Switched to branch 'conflict'
shashank@TTNPL-shashankpandey:~/test/demo$ git merge master
Auto-merging main.txt
CONFLICT (content): Merge conflict in main.txt
Automatic merge failed; fix conflicts and then commit the result.
shashank@TTNPL-shashankpandey:~/test/demo$
```

```
<<<<<< HEAD
heloo
=====
hiiii
>>>>>> master
~
```

```
Switched to branch 'conflict'
shashank@TTNPL-shashankpandey:~/test/demo$ git merge master
Auto-merging main.txt
CONFLICT (content): Merge conflict in main.txt
Automatic merge failed; fix conflicts and then commit the result.
shashank@TTNPL-shashankpandey:~/test/demo$ vim main.txt
shashank@TTNPL-shashankpandey:~/test/demo$ git merge master
error: Merging is not possible because you have unmerged files.
hint: Fix them up in the work tree, and then use 'git add/rm <file>'
hint: as appropriate to mark resolution and make a commit.
fatal: Exiting because of an unresolved conflict.
shashank@TTNPL-shashankpandey:~/test/demo$ git add .
shashank@TTNPL-shashankpandey:~/test/demo$ git merge master
fatal: You have not concluded your merge (MERGE_HEAD exists).
Please, commit your changes before you merge.
shashank@TTNPL-shashankpandey:~/test/demo$ git commit -m "resolve merge conflict"
[conflict 494c7ff] resolve merge conflict
shashank@TTNPL-shashankpandey:~/test/demo$ git merge master
Already up to date.
shashank@TTNPL-shashankpandey:~/test/demo$
```



Ans 17

```
[content is dirty] resolve merge conflict
shashank@TTNPL-shashankpandey:~/test/demo$ git merge master
Already up to date.
shashank@TTNPL-shashankpandey:~/test/demo$ echo "new change" >> main.txt
shashank@TTNPL-shashankpandey:~/test/demo$ git checkout -b p1
Switched to a new branch 'p1'
shashank@TTNPL-shashankpandey:~/test/demo$ git checkout main
error: pathspec 'main' did not match any file(s) known to git
shashank@TTNPL-shashankpandey:~/test/demo$ git checkout master
error: Your local changes to the following files would be overwritten by checkout:
    main.txt
Please commit your changes or stash them before you switch branches.
Aborting
shashank@TTNPL-shashankpandey:~/test/demo$ git stash
Saved working directory and index state WIP on p1: 494c7ff resolve merge conflict
shashank@TTNPL-shashankpandey:~/test/demo$ git checkout master
Switched to branch 'master'
shashank@TTNPL-shashankpandey:~/test/demo$ git stash pop
Auto-merging main.txt
CONFLICT (content): Merge conflict in main.txt
On branch master
Unmerged paths:
  (use "git restore --staged <file>..." to unstage)
  (use "git add <file>..." to mark resolution)
    both modified:   main.txt

no changes added to commit (use "git add" and/or "git commit -a")
The stash entry is kept in case you need it again.
shashank@TTNPL-shashankpandey:~/test/demo$ git add .
shashank@TTNPL-shashankpandey:~/test/demo$ git commit -m "updated main.txt"
[master b978de2] updated main.txt
 1 file changed, 5 insertions(+)
```

Ans 18 [Took help from chatgpt to understand]

```
shashank@TTNPL-shashankpandey:~$ vi .bashrc
shashank@TTNPL-shashankpandey:~$ source ./bashrc
bash: ./bashrc: No such file or directory
shashank@TTNPL-shashankpandey:~$ source ~/.bashrc
shashank@TTNPL-shashankpandey:~$ cd bootcamp-test-repo/
shashank@TTNPL-shashankpandey:bootcamp-test-repo (main)$
```