

Git Exercise

Q1. Git Setup

```
ishmeet-singh@TTNPL-ishmeet:~$ sudo apt install git
[sudo] password for ishmeet-singh:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
git is already the newest version (1:2.43.0-1ubuntu7.3).
The following packages were automatically installed and are no longer required:
  libgl1-amber-dri libglapi-mesa libllvm19
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 11 not upgraded.
ishmeet-singh@TTNPL-ishmeet:~$ git --version
git version 2.43.0
ishmeet-singh@TTNPL-ishmeet:~$ █
```

Q2. Initialize a Git Repository

```
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ 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/ishmeet-singh/git-exercise/.git/
```

Q3. Add files to the repository

```
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ echo "This is file1" >> file1.txt
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ echo "This is file2" >> file2.txt
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ echo "This is file3" >> file3.txt
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    file1.txt
    file2.txt
    file3.txt

nothing added to commit but untracked files present (use "git add" to track)
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ git add .
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   file1.txt
    new file:   file2.txt
    new file:   file3.txt

ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ █
```

Q4. Unstage 1 file

```
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ git reset file3.txt
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   file1.txt
    new file:   file2.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    file3.txt

ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ █
```

Q5. Commit the file

```
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ git commit -m "Add file1.txt and file.txt"
[master (root-commit) 5f9bae0] Add file1.txt and file.txt
 2 files changed, 2 insertions(+)
 create mode 100644 file1.txt
 create mode 100644 file2.txt
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ █
```

Q6. Add a remote

```
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ git branch
* master
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ git branch -M main
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ git remote add origin https://github.com/Ishmeet1202/git-exercise.git
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ █
```

Q7. Undo changes to a particular file

```
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ echo "This text should need to be undo" >> file1.txt
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ cat file1.txt
MAIN BRANCH CHANGED
file1 is changed from cloned repo
Main branch change after gitignore
FEATURE BRANCH CHANGED
file1 is changed from cloned repo
Feature branch change after gitignore
This text should need to be undo
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ git status
On branch main
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:   file1.txt

no changes added to commit (use "git add" and/or "git commit -a")
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ git restore file1.txt
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ git status
On branch main
nothing to commit, working tree clean
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ cat file1.txt
MAIN BRANCH CHANGED
file1 is changed from cloned repo
Main branch change after gitignore
FEATURE BRANCH CHANGED
file1 is changed from cloned repo
Feature branch change after gitignore
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ █
```

Q8. Push changes to Github.

```
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   file2.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    file3.txt

ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ git commit -m "Added changes to file2.txt"
[main 810c9b3] Added changes to file2.txt
 1 file changed, 1 insertion(+)
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ git push origin main
Username for 'https://github.com': Ishmeet1202
Password for 'https://Ishmeet1202@github.com':
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (7/7), 560 bytes | 560.00 KiB/s, done.
Total 7 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Ishmeet1202/git-exercise.git
 * [new branch]      main -> main
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ git status
On branch main
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    file3.txt

nothing added to commit but untracked files present (use "git add" to track)
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$
```

Q9. Clone the repository

```
ishmeet-singh@TTNPL-ishmeet:~/git-exercise-clone$ git clone https://github.com/Ishmeet1202/git-exercise.git
Cloning into 'git-exercise'...
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 7 (delta 0), reused 7 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (7/7), done.
ishmeet-singh@TTNPL-ishmeet:~/git-exercise-clone$ ls
git-exercise
ishmeet-singh@TTNPL-ishmeet:~/git-exercise-clone$
```

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

```
ishmeet-singh@TTNPL-ishmeet:~/git-exercise-clone/git-exercise$ echo "file1 is changed from cloned repo" >> file1.txt
ishmeet-singh@TTNPL-ishmeet:~/git-exercise-clone/git-exercise$ git status
On branch main
Your branch is up to date with 'origin/main'.

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:   file1.txt

no changes added to commit (use "git add" and/or "git commit -a")
ishmeet-singh@TTNPL-ishmeet:~/git-exercise-clone/git-exercise$ git commit -m "Modify file1 from clonned repo"
On branch main
Your branch is up to date with 'origin/main'.

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:   file1.txt

no changes added to commit (use "git add" and/or "git commit -a")
ishmeet-singh@TTNPL-ishmeet:~/git-exercise-clone/git-exercise$ git add file1.txt
ishmeet-singh@TTNPL-ishmeet:~/git-exercise-clone/git-exercise$ git commit -m "Modify file1 from clonned repo"
[main b21d56b] Modify file1 from clonned repo
 1 file changed, 1 insertion(+)
ishmeet-singh@TTNPL-ishmeet:~/git-exercise-clone/git-exercise$ git push origin main
Username for 'https://github.com': Ishmeet1202
Password for 'https://Ishmeet1202@github.com':
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 326 bytes | 326.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Ishmeet1202/git-exercise.git
  810c9b3..b21d56b  main -> main
ishmeet-singh@TTNPL-ishmeet:~/git-exercise-clone/git-exercise$
```

```
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ git pull origin main
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 306 bytes | 306.00 KiB/s, done.
From https://github.com/Ishmeet1202/git-exercise
 * branch            main      -> FETCH_HEAD
   810c9b3..b21d56b  main      -> origin/main
Updating 810c9b3..b21d56b
Fast-forward
 file1.txt | 1 +
 1 file changed, 1 insertion(+)
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ cat file1.txt
This is file1
file1 is changed from cloned repo
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$
```

Q11. Check differences between a file and its staged version

```
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ git status
On branch main
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    file3.txt

nothing added to commit but untracked files present (use "git add" to track)
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ echo "file2.txt is modified" >> file2.txt
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ git status
On branch main
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:   file2.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    file3.txt

no changes added to commit (use "git add" and/or "git commit -a")
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ git diff
diff --git a/file2.txt b/file2.txt
index 095ee29..ef26cd8 100644
--- a/file2.txt
+++ b/file2.txt
@@ -1 +1,2 @@
 This is file2
+file2.txt is modified
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ git add file2.txt
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ git diff --staged
diff --git a/file2.txt b/file2.txt
index 095ee29..ef26cd8 100644
--- a/file2.txt
+++ b/file2.txt
@@ -1 +1,2 @@
 This is file2
+file2.txt is modified
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ █
```

Q12. Ignore a few files to be checked in

```
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ git status
On branch main
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    file3.txt

nothing added to commit but untracked files present (use "git add" to track)
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ nano .gitignore
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ git status
On branch main
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .gitignore

nothing added to commit but untracked files present (use "git add" to track)
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ git add .
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ git commit -m "Adding gitignore file"
[main a5d5cf9] Adding gitignore file
 1 file changed, 1 insertion(+)
  create mode 100644 .gitignore
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ git push origin main
Username for 'https://github.com': Ishmeet1202
Password for 'https://Ishmeet1202@github.com':
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), 319 bytes | 319.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Ishmeet1202/git-exercise.git
  b21d56b..a5d5cf9  main -> main
```

Q13. Create a new branch

```
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ git branch
* main
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ git branch feature
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ git branch
  feature
* main
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ git checkout feature
Switched to branch 'feature'
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$
```

Q14. Diverge them with commits

```
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ git checkout main
Already on 'main'
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ echo "Main branch change after gitignore" >> file1.txt
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ git add file1.txt
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ git commit -m "Main branch commit after gitignore" [main 572f075] Main branch commit after gitignore
 1 file changed, 1 insertion(+)
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ git checkout feature
Switched to branch 'feature'
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ echo "Feature branch change after gitignore" >> file1.txt
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ git add file1.txt
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ git commit -m "Feature branch commit after gitignore" [feature cb5a9b0] Feature branch commit after gitignore
 1 file changed, 1 insertion(+)
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ git status
On branch feature
nothing to commit, working tree clean
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ git log --oneline --graph --all
* cb5a9b0 (HEAD -> feature) Feature branch commit after gitignore
| * 572f075 (main) Main branch commit after gitignore
||
* a5d5cf9 (origin/main) Adding gitignore file
* b21d56b Modify file1 from clonned repo
* 810c9b3 Added changes to file2.txt
* 5f9bae0 Add file1.txt and file.txt
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$
```

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

```
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ git checkout main
Switched to branch 'main'
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ nano file1.txt
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ git add file1.txt
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ git commit -m "Edit same line in file1 on main" [main fd9c4be] Edit same line in file1 on main
 1 file changed, 1 insertion(+), 1 deletion(-)
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ git checkout feature
Switched to branch 'feature'
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ nano file1.txt
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ git add file1.txt
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ git commit -m "Edit same line in file1 on feature" [feature bba875c] Edit same line in file1 on feature
 1 file changed, 1 insertion(+), 1 deletion(-)
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ git log --oneline --graph --all
* bba875c (HEAD -> feature) Edit same line in file1 on feature
* cb5a9b0 Feature branch commit after gitignore
| * fd9c4be (main) Edit same line in file1 on main
| * 572f075 Main branch commit after gitignore
||
* a5d5cf9 (origin/main) Adding gitignore file
* b21d56b Modify file1 from clonned repo
* 810c9b3 Added changes to file2.txt
* 5f9bae0 Add file1.txt and file.txt
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$
```

The screenshot shows a Linux desktop environment with a terminal window and a file manager window.

Terminal Window:

- Terminal title: ishmeet:singh@TTNPL:ishmeet: ~
- Terminal content:

```
GNU nano 7.2
MAIN BRANCH CHANGED
file1 is changed from cloned repo
Main branch change after gitignore
```
- Terminal status bar: file1.txt

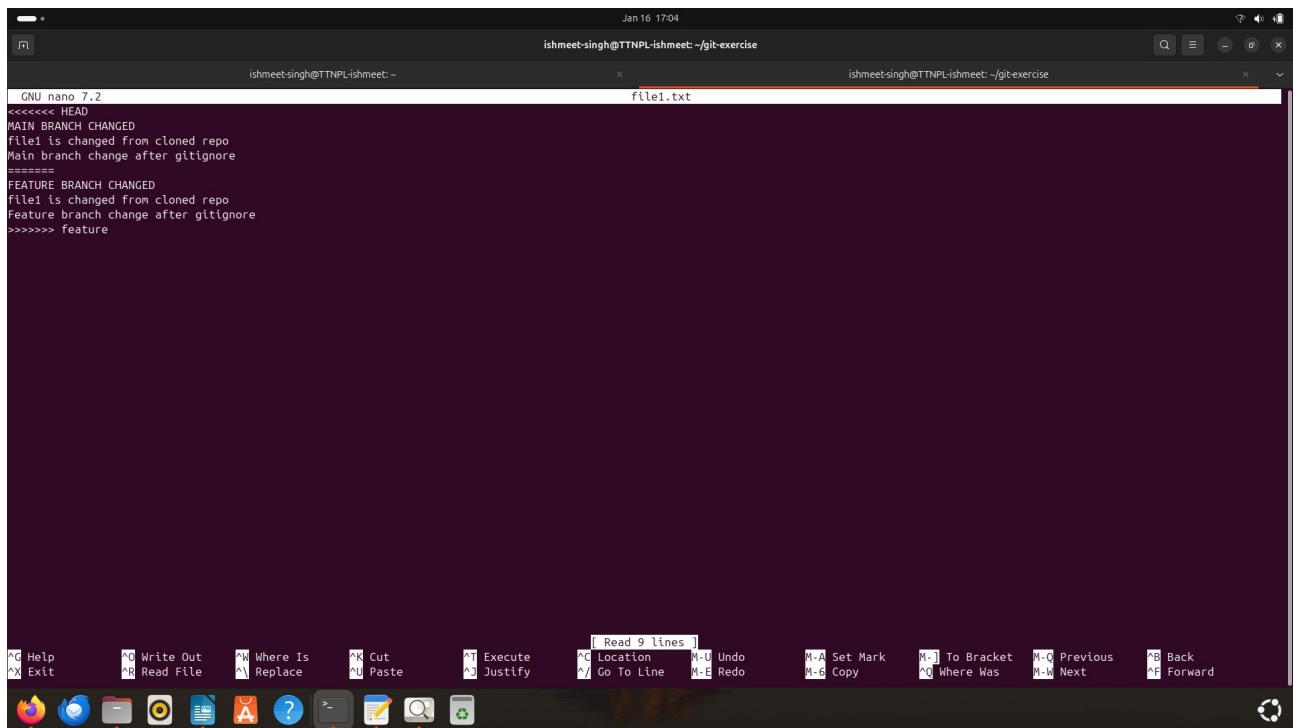
File Manager Window:

- Window title: file1.txt
- Content: [Read 3 lines]
- File operations menu:
 - [R] Read File
 - [W] Write Out
 - [A] Replace
 - [C] Cut
 - [P] Paste
 - [E] Execute
 - [J] Justify
 - [L] Location
 - [G] Go To Line
 - [U] Undo
 - [R] Redo
 - [S] Set Mark
 - [B] To Bracket
 - [W] Where Was
 - [P] Previous
 - [N] Next
 - [B] Back
 - [F] Forward

A screenshot of a Linux desktop environment. At the top, there's a dark header bar with the date "Jan 16 16:54" and a user "ishmeet.singh@TTNPL-ishmeet". Below the header, there are two terminal windows. The left window shows the output of "git status" with messages like "FEATURE BRANCH CHANGED", "file1 is changed from cloned repo", and "Feature branch change after gitignore". The right window is titled "file1.txt" and is currently empty. A horizontal menu bar is visible above the desktop, containing icons for Help, Exit, Write Out, Read File, Where Is, Replace, Cut, Paste, Execute, Justify, Location, Undo, Redo, Set Mark, To Bracket, Previous, Next, Back, and Forward. At the bottom, a docked application bar features icons for various desktop applications, including a browser, file manager, terminal, and system settings.

Q16. Try merging and resolve merge conflicts

```
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ git checkout main
Already on 'main'
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ git merge feature
Auto-merging file1.txt
CONFLICT (content): Merge conflict in file1.txt
Automatic merge failed; fix conflicts and then commit the result.
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ nano file1.txt
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ git add file1.txt
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ git commit -m "Resolve merge conflict between main and feature"
[main 2a93741] Resolve merge conflict between main and feature
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ git status
On branch main
nothing to commit, working tree clean
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ git log --oneline --graph --all
* 2a93741 (HEAD -> main) Resolve merge conflict between main and feature
|\ \
| * bba875c (feature) Edit same line in file1 on feature
| * cb5a9b0 Feature branch commit after gitignore
* | fd9cabe Edit same line in file1 on main
* | 572f075 Main branch commit after gitignore
|/
* a5d5cf9 (origin/main) Adding gitignore file
* b21d56b Modify file1 from cloned repo
* 810c9b3 Added changes to file2.txt
* 5f9bae0 Add file1.txt and file.txt
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$
```



```
ishmeet-singh@TTNPL-ishmeet:~
```

```
file1.txt
```

```
ishmeet-singh@TTNPL-ishmeet:~/git-exercise
```

```
GNU nano 7.2
```

```
<<<<< HEAD
MAIN BRANCH CHANGED
file1 is changed from cloned repo
Main branch change after gitignore
=====
FEATURE BRANCH CHANGED
file1 is changed from cloned repo
Feature branch change after gitignore
>>>>> feature
```

```
^G Help      ^O Write Out   ^W Where Is   ^X Cut        ^T Execute   [ Read 9 lines ]
^X Exit      ^P Read File   ^R Replace    ^U Paste     ^I Justify   ^C Location   M-U Undo
                                         M-D Redo    M-A Set Mark M-B To Bracket M-Q Previous M-H Next
                                         M-G Copy    M-Q Where Was M-F Back   M-F Forward
```

Q17. Stash the changes and pop them

```
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ git checkout main
Already on 'main'
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ echo "This change will be stashed" >> file1.txt
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ git status
On branch main
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:   file1.txt

no changes added to commit (use "git add" and/or "git commit -a")
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ git stash
Saved working directory and index state WIP on main: 2a93741 Resolve merge conflict between main and feature
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ git stash pop
On branch main
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:   file1.txt

no changes added to commit (use "git add" and/or "git commit -a")
Dropped refs/stash@{0} (d2c5dd4a6ba68f44a177eeca79fbdf942057a18)
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$
```

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_prompt
```

```
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ git checkout main
Already on 'main'
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ nano ~/.bashrc
ishmeet-singh@TTNPL-ishmeet:~/git-exercise$ source ~/.bashrc
ishmeet-singh@TTNPL-ishmeet:git-exercise (main)$ git checkout feature
Switched to branch 'feature'
ishmeet-singh@TTNPL-ishmeet:git-exercise (feature)$
```

```
Jan 16 17:31 ishmeet-singh@TTNPL-ishmeet: ~/git-exercise
ishmeet-singh@TTNPL-ishmeet: ~/git-exercise
GNU nano 7.2 /home/ishmeet-singh/.bashrc *
alias alert='notify-send --urgency=low -i "$(($? == 0) && echo terminal || echo error)" "$(history|tail -1|sed -e '\\'$'^{0-9}\)+\s*//;s/[;8|]\s*alerts//\'')"'*
# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.
if [ -f ~/.bash_aliases ]; then
. ~/.bash_aliases
fi
# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -q posix; then
if [ -f /usr/share/bash-completion/bash_completion ]; then
. /usr/share/bash-completion/bash_completion
elif [ -f /etc/bash_completion ]; then
. /etc/bash_completion
fi
fi
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
^C Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo M-A Set Mark M-B To Bracket M-Q Previous ^B Back
^X Exit ^R Read File ^A Replace ^P Paste ^J Justify ^G Go To Line M-E Redo M-G Copy M-Q Where Was M-W Next ^F Forward
```