**Exercise 1: Create your first Commit.**

**Task:**

*Initialize your Repository: git init project Exercise-1*

*Create a file with name index.html*

*Write code in newly created file*

*Stage index.html*

*Commit with message ‘Initial Message.*’

Command:

1.cd Exercise1

2. git init

3.vi index.html

Esc, shift+:,x

4.git add .

5.git commit -m”Initial Message”

**Exercise 2: Commit only one file.**

**Task:**

* *Initialize your Repository: git init Exercise-2*
* *Create a files with names index.html and about.html*
* *Add code in both files. Stage only one file.*
* *Commit with message: ‘File x added.’*

Command:

1.cd Exercise2

2. git init

3.vi index.html

Esc, shift+:,x

4. vi about.html

Esc, shift+:,x

5.git add .

6.git commit -m”Index.html file added”

**Exercise 3: Modify last commit.**

**Task:**

* *Initialize your Repository: git init project Exercise-3*
* *Create a file with name index.html*
* *Write code in newly created file Stage index.html*
* *Commit with message ‘adding index.html.’*
* *Add new about.html file with some code.*
* *Now modify message of previous commit as ‘Adding index.html and about.html ’and add about.html file in previous commit.*

Command:

1.cd Exercise3

2. git init

3.vi index.html

Esc, shift+:,x

4.git add .

5. git commit -m”adding index.html”

6. vi about.html

Esc, shift+:,x

7. git commit –amend

8.git status

9. git lola

Exercise 4: Create git ignore file

**Task:**

* *Initialize your Repository: git init Exercise-4*
* *Create a gitignore file that will ignore*
* *all files with ‘exe’ extension*
* *all files with ‘o’ extension*
* *all files with ‘jar’ extension*
* *the whole libraries directory*

command:

1.cd Exercise4

2. git init

3.vi. gitignore

Code- \*.exe,\*.o,\*.jar,/libraries

Esc+shitt+:,x

**Exercise 5: Save your work with stash.**

**Task:**

* *Initialize your Repository: git init Exercise-5*
* *Add some random files with code*
* *Now use git stash command to save required files from above files.*

Command:

1.cd Exercise5

2. git init

3.vi index.html

Esc, shift+:,x

4.git add .

5.git commit -m”added index.html file”

6. vi me.html

Esc, shift+:,x

7. vi profile.html

Esc, shift+:,x

8. git stash

9.git status

10 git stash show

11.git stash lola

**Exercise 6: Reset Previous Commit Keeping changes. (Soft Reset)**

**Task:**

* *Initialize your Repository: git init Exercise-6.*
* *Do some series of commits.*
* *Now use git reset command to reset one of the commit and keep changes in staging area.*

*Command:*

1.cd Exercise6

2. git init

3.vi index.html

Esc, shift+:,x

4.git add .

5.git commit -m”added index.html file”

6. vi me.html

Esc, shift+:,x

7.git commit -m”added me.html file”

8 git reset head~1

9.git status

Exercise 7: Reset Previous Commit (Hard Reset).

**Task:**

* *Initialize your Repository: git init Exercise-7.*
* *Do some series of commits.*
* *Use git reset command to reset one of the commit without keeping changes.*

*Command:*

1.cd Exercise7

2. git init

3.vi index.html

Esc, shift+:,x

4.git add .

5.git commit -m”added index.html file”

6. vi me.html

Esc, shift+:,x

7.git commit -m”added me.html file”

8. git reset head~1 –hard

9.git status (output-On branch master

nothing to commit, working tree clean)

**Exercise 8: Merge branch (No Conflicts).**

**Task:**

* *Initialize your Repository: git init Exercise-8*
* *Create index.html file add some code.*
* *Make Commit.*
* *Create new branch from master branch with name my-feature.*
* *Add about-us.html file with code and commit changes in my-feature branch.*
* *Now checkout master branch.*
* *Merge my-feature branch into master*

*Command:*

1.cd Exercise8

2. git init

3.vi index.html

Esc, shift+:,x

4.git add .

5.git commit -m”added index.html file”

6.git branch my\_feature

7. vi index.html

Esc, shift+:,x

8.git add .

9.git commit -m”added index.html”

10 git checkout master

11. git merge my\_feature

**Exercise 9: Merge branch (Conflicts).**

**Task:**

* *Initialize your Repository: git init Exercise-9*
* *Create index.html file add some code.*
* *Make Commit.*
* *Create new branch from master branch with name my-feature.*
* *Add about-us.html file with code*
* *Make changes in index.html*
* *Commit changes in my-feature branch.*
* *Now checkout master branch.*
* *Do some code changes in index.html Commit changes?*
* *Now merge my-feature branch into master.*
* *You will get conflicts for index.html file.Resolve conflict by selecting both changes for the same index.html file*

*Command:*

1.cd Exercise8

2. git init

3.vi index.html ,code: reeti bhagat profile

Esc, shift+:,x

4.git add .

5.git commit -m”added index.html file”

6.git branch my-feature

7. vi about.html

Esc, shift+:,x

8.vi index.html code: new line added

Esc, shift+:,x

9.git add .

10 git commit -m” changed made”

11.git checkout master

12 vi index.html code: added second line

Esc +shift+:,x

13. git merge my-feature

14 vi index.html, code: make changes  
esc,shift+:,x

15.git status

16 git lola