

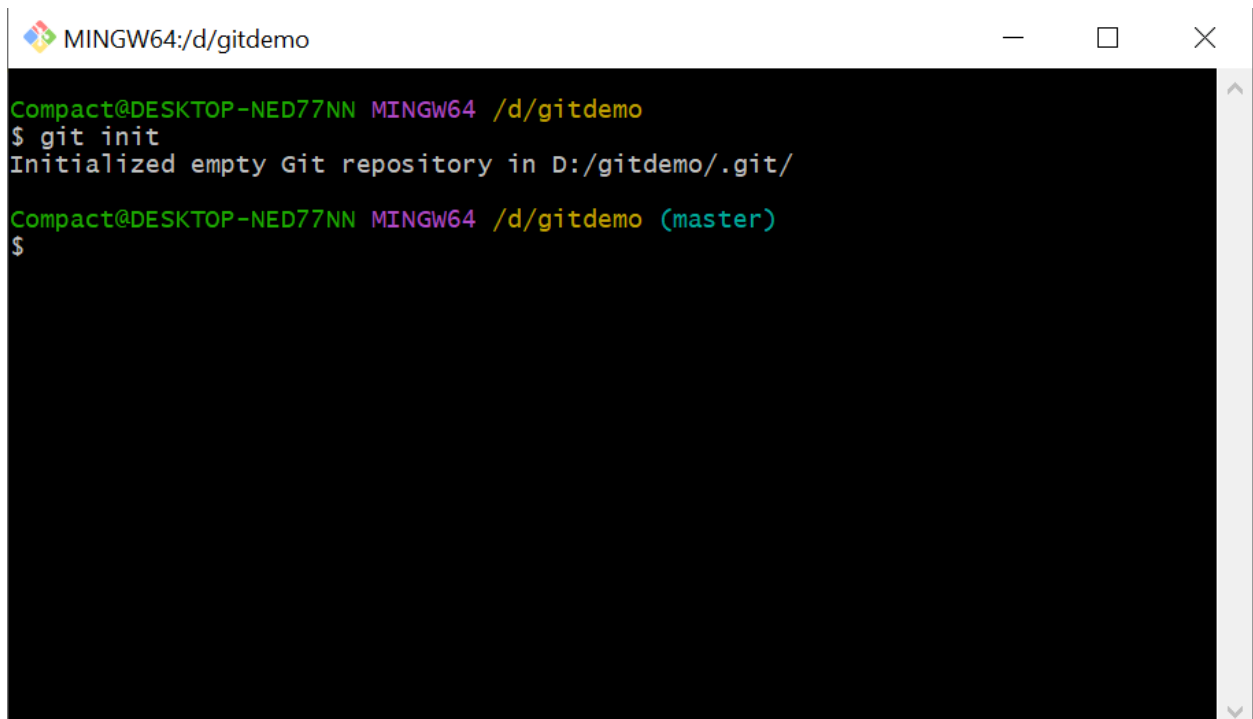
ASSIGNMENT-1

Assignment 1: Initialize a new Git repository in a directory of your choice. Add a simple text

file to the repository and make the first commit.

Step1: Initialize the repository:

➤ Run the git init command in the terminal to create a new Git repository.

A screenshot of a Windows terminal window titled 'MINGW64:/d/gitdemo'. The window has standard Windows window controls (minimize, maximize, close) in the top right corner. The terminal shows the following text: 'Compact@DESKTOP-NED77NN MINGW64 /d/gitdemo' on the first line, '\$ git init' on the second line, and 'Initialized empty Git repository in D:/gitdemo/.git/' on the third line. The prompt '\$' appears again on the fourth line, indicating the command was successful. The terminal background is black, and the text is in a light green/cyan color. A vertical scrollbar is visible on the right side of the terminal window.

```
MINGW64:/d/gitdemo
Compact@DESKTOP-NED77NN MINGW64 /d/gitdemo
$ git init
Initialized empty Git repository in D:/gitdemo/.git/
Compact@DESKTOP-NED77NN MINGW64 /d/gitdemo (master)
$
```

Step 2: Create a Simple Text File

➤ echo command outputs the string "This is my first commit" and the > operator redirects this

output into a new file named README.txt.

 MINGW64:/d/gitdemo

```
Compact@DESKTOP-NED77NN MINGW64 /d/gitdemo (master)
$ git init
Reinitialized existing Git repository in D:/gitdemo/.git
Compact@DESKTOP-NED77NN MINGW64 /d/gitdemo (master)
$ echo "this is my first commit" >README.txt
Compact@DESKTOP-NED77NN MINGW64 /d/gitdemo (master)
$ |
```

 helloworld.txt - Notepad

File Edit Format View Help

this is your hello world

```
MINGW64:/d/gitdemo

Compact@DESKTOP-NED77NN MINGW64 /d/gitdemo (master)
$ git init
Reinitialized existing Git repository in D:/gitdemo/.git/

Compact@DESKTOP-NED77NN MINGW64 /d/gitdemo (master)
$ echo "this is my first commit" >README.txt

Compact@DESKTOP-NED77NN MINGW64 /d/gitdemo (master)
$ git add README.txt
warning: in the working copy of 'README.txt', LF will be replaced by CRLF the next time Git touches it

Compact@DESKTOP-NED77NN MINGW64 /d/gitdemo (master)
$ |
```

Step 3: Add the File to the Staging

```
MINGW64:/d/gitdemo

$ echo "this is my first commit" >README.txt

Compact@DESKTOP-NED77NN MINGW64 /d/gitdemo (master)
$ git add README.txt
warning: in the working copy of 'README.txt', LF will be replaced by CRLF the next time Git touches it

Compact@DESKTOP-NED77NN MINGW64 /d/gitdemo (master)
$ git add .

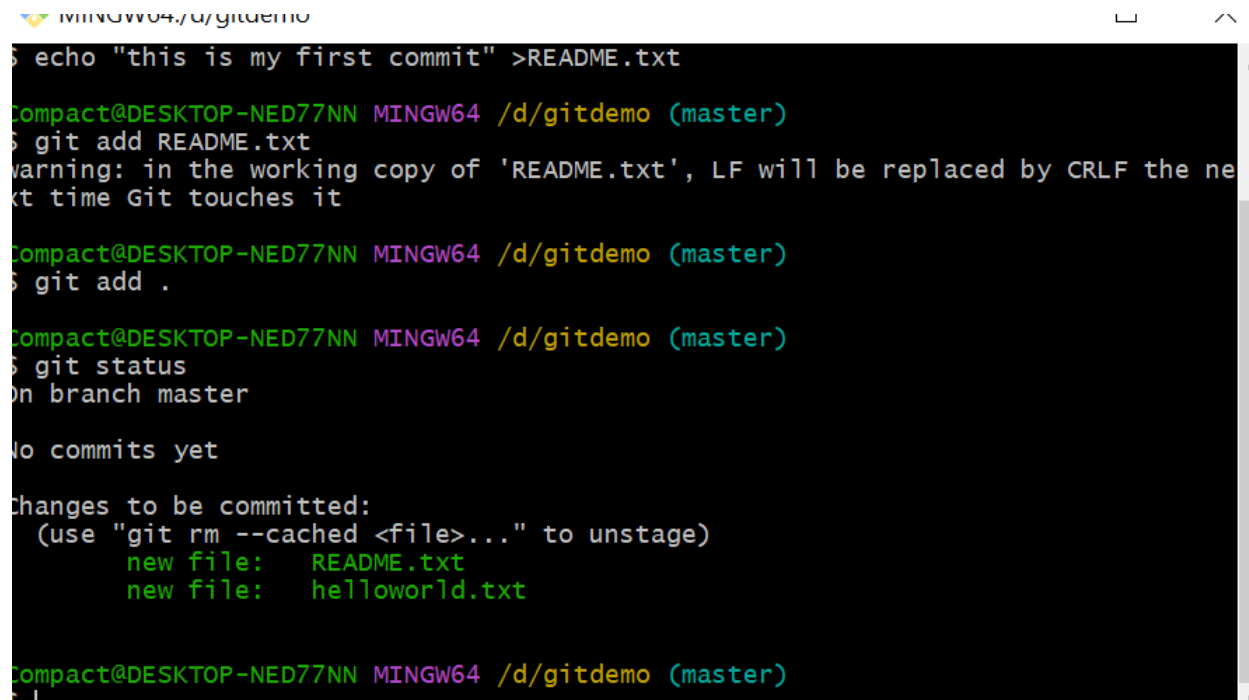
Compact@DESKTOP-NED77NN MINGW64 /d/gitdemo (master)
$ git status
On branch master

no commits yet

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

Compact@DESKTOP-NED77NN MINGW64 /d/gitdemo (master)
$ |
```

Step 5: Now here added the file

A screenshot of a Windows command prompt window titled 'MINGW64 /d/gitdemo'. The terminal shows the following sequence of commands and output: 1. Command: 'echo "this is my first commit" >README.txt'. 2. Command: 'git add README.txt'. Output: 'warning: in the working copy of 'README.txt', LF will be replaced by CRLF the next time Git touches it'. 3. Command: 'git add .'. 4. Command: 'git status'. Output: 'On branch master', 'No commits yet', 'Changes to be committed: (use "git rm --cached <file>..." to unstage)', 'new file: README.txt', 'new file: helloworld.txt'. 5. Command: 'git commit -m "Initial commit with README.txt"'. The prompt 'Compact@DESKTOP-NED77NN MINGW64 /d/gitdemo (master)' is visible at the start of each command line.

```
Compact@DESKTOP-NED77NN MINGW64 /d/gitdemo (master)
$ echo "this is my first commit" >README.txt
Compact@DESKTOP-NED77NN MINGW64 /d/gitdemo (master)
$ git add README.txt
warning: in the working copy of 'README.txt', LF will be replaced by CRLF the next time Git touches it
Compact@DESKTOP-NED77NN MINGW64 /d/gitdemo (master)
$ git add .
Compact@DESKTOP-NED77NN MINGW64 /d/gitdemo (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   README.txt
        new file:   helloworld.txt
Compact@DESKTOP-NED77NN MINGW64 /d/gitdemo (master)
$ git commit -m "Initial commit with README.txt"
```

Step 5: Commit the Changes

➤ The git commit command creates a new commit from the changes in the staging area. The -m

flag allows you to add a commit message, in this case, "Initial commit with README.txt"

MINGW64:/d/gitdemo

```
compact@DESKTOP-NED77NN MINGW64 /d/gitdemo (master)
$ git add .

compact@DESKTOP-NED77NN MINGW64 /d/gitdemo (master)
$ git status
On branch master

no commits yet

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

compact@DESKTOP-NED77NN MINGW64 /d/gitdemo (master)
$ git commit -m "this is my first commit"
[master (root-commit) a10396d] this is my first commit
2 files changed, 2 insertions(+)
create mode 100644 README.txt
create mode 100644 helloworld.txt

compact@DESKTOP-NED77NN MINGW64 /d/gitdemo (master)
```

Step 6: List All Branches

➤ The git branch command lists all the branches in the repository. The * indicates the current branch, which is master.

```
MINGW64:/d/gitdemo
$ git status
On branch master

No commits yet

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


Compact@DESKTOP-NED77NN MINGW64 /d/gitdemo (master)
$ git commit -m "this is my first commit"
[master (root-commit) a10396d] this is my first commit
 2 files changed, 2 insertions(+)
 create mode 100644 README.txt
 create mode 100644 helloworld.txt

Compact@DESKTOP-NED77NN MINGW64 /d/gitdemo (master)
$ git branch
* master

Compact@DESKTOP-NED77NN MINGW64 /d/gitdemo (master)
$
```

Step 7: Create a New Branch

The git branch feature command creates a new branch named feature.

 MINGW64:/d/gitdemo

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

Compact@DESKTOP-NED77NN MINGW64 /d/gitdemo (master)
$ git commit -m "this is my first commit"
[master (root-commit) a10396d] this is my first commit
 2 files changed, 2 insertions(+)
 create mode 100644 README.txt
 create mode 100644 helloworld.txt

Compact@DESKTOP-NED77NN MINGW64 /d/gitdemo (master)
$ git branch
* master


Compact@DESKTOP-NED77NN MINGW64 /d/gitdemo (master)
$ git checkout -b sahbaz
Switched to a new branch 'sahbaz'

Compact@DESKTOP-NED77NN MINGW64 /d/gitdemo (sahbaz)
$ |
```

Step 8: Switch to the New Branch

➤ The git checkout feature command switches the current working branch to feature. This

means any new commits will be made on the feature branch.

 MINGW64:/d/gitdemo

new file: helloworld.txt

```
Compact@DESKTOP-NED77NN MINGW64 /d/gitdemo (master)
$ git commit -m "this is my first commit"
[master (root-commit) a10396d] this is my first commit
2 files changed, 2 insertions(+)
create mode 100644 README.txt
create mode 100644 helloworld.txt

Compact@DESKTOP-NED77NN MINGW64 /d/gitdemo (master)
$ git branch
* master

Compact@DESKTOP-NED77NN MINGW64 /d/gitdemo (master)
$ git checkout -b sahbaz
Switched to a new branch 'sahbaz'


Compact@DESKTOP-NED77NN MINGW64 /d/gitdemo (sahbaz)
$ git checkout sahbaz
Already on 'sahbaz'

Compact@DESKTOP-NED77NN MINGW64 /d/gitdemo (sahbaz)
$
```

Step 9: Verify the Branch Switch

➤ Running git branch again lists all branches and shows that feature is now the current branch

(indicated by the *).

 MINGW64:/d/gitdemo

```
[master (root-commit) a10396d] this is my first commit
2 files changed, 2 insertions(+)
create mode 100644 README.txt
create mode 100644 helloworld.txt
```

```
Compact@DESKTOP-NED77NN MINGW64 /d/gitdemo (master)
$ git branch
* master
```

```
Compact@DESKTOP-NED77NN MINGW64 /d/gitdemo (master)
$ git checkout -b sahbaz
Switched to a new branch 'sahbaz'
```

```
Compact@DESKTOP-NED77NN MINGW64 /d/gitdemo (sahbaz)
$ git checkout sahbaz
Already on 'sahbaz'
```

```
Compact@DESKTOP-NED77NN MINGW64 /d/gitdemo (sahbaz)
$ git branch
  master
* sahbaz
```

```
Compact@DESKTOP-NED77NN MINGW64 /d/gitdemo (sahbaz)
$
```