

```
Student@MACE-29 MINGW64 ~ (master)
$ mkdir GitLab-1HC24CS058

Student@MACE-29 MINGW64 ~ (master)
$ cd GitLab-1HC24CS058

Student@MACE-29 MINGW64 ~/GitLab-1HC24CS058 (master)
$ git init
Initialized empty git repository in C:/Users/student/GitLab-1HC24CS058/.git/

Student@MACE-29 MINGW64 ~/GitLab-1HC24CS058 (master)
$ git config --global user.name "prajwal-1HC24CS058"

Student@MACE-29 MINGW64 ~/GitLab-1HC24CS058 (master)
$ git config --global user.email "prajwal158@gmail.com"

Student@MACE-29 MINGW64 ~/GitLab-1HC24CS058 (master)
$ echo "GitLab Program 1">gitlab-prgm-1.txt

Student@MACE-29 MINGW64 ~/GitLab-1HC24CS058 (master)
$ cat gitlab-prgm-1.txt
GitLab Program 1

Student@MACE-29 MINGW64 ~/GitLab-1HC24CS058 (master)
$ git add gitlab-prgm-1.txt
warning: in the working copy of 'gitlab-prgm-1.txt', LF will be replaced by CRLF the next time git touches it

Student@MACE-29 MINGW64 ~/GitLab-1HC24CS058 (master)
$ git add gitlab-prgm-1.txt

Student@MACE-29 MINGW64 ~/GitLab-1HC24CS058 (master)
$ git status
On branch master

no commits yet

changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   gitlab-prgm-1.txt

Student@MACE-29 MINGW64 ~/GitLab-1HC24CS058 (master)
$ git commit -m "add gitlab-prgm-1.txt in master branch"
[master (root-commit) e262bd1] add gitlab-prgm-1.txt in master branch
 1 file changed, 1 insertion(+)
 create mode 100644 gitlab-prgm-1.txt

Student@MACE-29 MINGW64 ~/GitLab-1HC24CS058 (master)
$ git status
On branch master
nothing to commit, working tree clean

Student@MACE-29 MINGW64 ~/GitLab-1HC24CS058 (master)
$ git log
commit e262bd1031fa35775d7862f0f171ee6887253b53 (HEAD -> master)
Author: Prajwal-1HC24CS058 <prajwal158@gmail.com>
Date: Thu Oct 30 14:38:40 2025 +0530

    add gitlab-prgm-1.txt in master branch
```

24

Initialised empty git repository in C:\Users\Student\gitlab-
Student@HNCE-29 MINGW64 / (master) 1hr success / git

On branch master

No commits yet

Changes to be committed:

(use "git rm --cached <file>..." to unstage)

new file: git-prgm-1.txt

[master (root-commit) e262bd1] add gitlab-prgm-1.txt in master branch
1 file changed, 1 insertion(+)
create mode 100644 gitlab-prgm-1.txt

Experiment Number: 1

Date: 11/10/2025

Setting Up and Basic Commands

Initialize a new git repository in a directory, create a new file and add it to the staging area and commit the changes with an appropriate commit message

Solution:

Step 1: To initialize git repository in a directory
\$ git init

Step 2: Create a new file and add it to staging area

\$ git add git-prgm-1.txt

To check status of git repository

\$ git status

Step 3: Commit the changes with an appropriate commit message. Replace "your commit message here" with meaningful description of your changes

\$ git commit -m "add git-prgm-1.txt into master branch"

If author is not identified, have to configure email, id

\$ git config --global user.name "Pragwal-1HC24CS058"

\$ git config --global user.email "pragwal58@gmail.com"

10/10

85
10/10/25

Already on 'master'

switched to a new branch 'feature-branch'
Student@HUNCE-29 MINGW64 ~/14gitlab-1HC2-1CSOS8 (feature-branch)



[feature-branch] 7:45:56 I added gitlab-fragm-2.txt to feature-branch
1 file changed, 1 inserted (+)
create mode 100644 gitlab-fragm-2.txt

Switched to branch 'master'

Student@HUNCE-29 MINGW64 ~/14gitlab-1HC2-1CSOS8 (master)

Experiment Number: 2

Date: 16/10/25

5

Creating and Managing Branches

Create a new branch named "feature-branch" switch to the "master" branch. Merge "feature-branch" into master.

Solution

Step 1: Make sure that you are in master branch

```
$ git checkout master
```

Step 2: Create a new branch named "feature-branch" and switch to it

```
$ git checkout -b feature-branch
```

Step 3: Make some changes in feature-branch by adding/modifying/deleting files as needed.

```
$ echo "GitLab Program No 2" > gitlab-prgm-2.txt
```

```
$ git add gitlab-prgm-2.txt
```

Step 4: Stage and commit your changes in the feature branch

```
$ git commit -m "added gitlab-prgm-2 to feature branch"
```

Step 5: Switch back to master branch

```
$ git checkout master
```

Step 6: Merge the "feature-branch" into the "master" branch

Updating e262bd1... 4748a5f

fast forward

gillab-prgm-2.txt 11 +

1 file changed, 1 insertion(+)

create mode 100644 gillab-prgm-2.txt

commit 4748a5f71... (HEAD -> master, feature-branch)

Author: Prajwal-HC26CS058 <prajwal58@gmail.com>

Date: Thu Oct 30 15:09:43 2025 +0530

added gillab-prgm-2 to feature branch

commit e262bd1031fa

Author: Prajwal-HC26CS058 <prajwal58@gmail.com>

Date: Thu Oct 30 16:38:40 2025 +0530

added gillab-prgm-1.txt in master branch.

Updating e2b2bd1...6748a5f

*fast forward

gitlab-prgm-2.txt 11 +

1 file changed, 1 insertion(+)

create mode 100644 gitlab-prgm-2.txt

commit 4748a5f71... (HEAD -> master, feature-branch)

Author: Prajwal-1HC24CS058 <prajwal58@gmail.com>

Date: Thu Oct 30 15:09:43 2025 +0530

added gitlab-prgm-2 to feature branch

commit e2b2bd1031fa

Author: Prajwal-1HC24CS058 <prajwal58@gmail.com>

Date: Thu Oct 30 14:38:40 2025 +0530

added gitlab-prgm-1.txt in master branch.

Experiment Number : Date :

\$git merge feature-branch

No check logs.

\$git log



83
10/11/25

No local changes to save

saved working directory and index stage on master: stash
temporary storage on master branch

on branch master

changes to be committed:

(use 'git restore --staged <file>...' to unstage)

new file: qitlab-prgm-3.txt

^ qitlab-prgm-3.txt

switched to branch feature-branch

Student@HNNCE-29 MINGW64 ~/qitlab-1HC24CS058 (feature-branch)

Experiment Number

3

Date

23/10/2025

7

Creating and Managing Branches

Write the command to stash changes, switch branches and then apply the stashed changes

stash - temporarily saves your changes to a stash so we can switch branches without committing or losing work

Solution: To stash changes, switch branches and then apply stashed changes in git, following commands are used

Step 1: Stash your changes. `$ git stash save "your message"`
`$ git stash save "stash temporary storage"`

Create and modify and then add a file using git add command called "gitlab-prgm-3.txt"
`$ git add gitlab-prgm-3.txt`

`$ git stash save "stash temporary storage" on master branch`

Apply the stash
`$ git stash apply`

Step 2: Switch to the desired branch
The desired branch here being feature branch
`$ git checkout feature-branch`

saved working directory and index stage on feature-branch
stash temporary storage on feature-branch.

On branch feature-branch:

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

new file: gitlab-frgm-3.txt

On branch feature-branch

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

new file: gitlab-frgm-3.txt

[feature-branch 029b82b] add gitlab-frgm-3.txt to feature ^{branch}
1 file changed, 1 insertion (+)
create mode 100644 gitlab-frgm-3.txt

commit 029b82b

HEAD -> feature-branch

Author: Prajwal-1HC24CS058 <prajwal58@gmail.com>

Date: Thu Oct 30 15:44:01 2025 +0530

add gitlab-frgm-3.txt to the feature branch.

Experiment Number _____ Date _____

Step 3: Apply the stashed changes

\$ git stash save "stash temporary storage on feature branch"

\$ git stash apply

To check status

\$ git status

Since git-prgm-3.txt is not committed to git repository, it is not visible in the repository logs. So to commit \$ git commit -m "add git-prgm-3.txt to feature branch"

To check logs

\$ git log



8/10/13