

```
student@HNUCE-29:~/gitlab-1hc24cs058$ git add gitlab-prgm-1.txt
student@HNUCE-29:~/gitlab-1hc24cs058$ git commit -m "add gitlab-prgm-1.txt in master branch"
[student@HNUCE-29:~/gitlab-1hc24cs058] e262bd1031fa35775d7802f0f171ee6887253b93 (HEAD -> master)
Author: Prajwal-1hc24cs058 <prajwal58@gmail.com>
Date:   Thu Oct 30 14:38:40 2025 +0530

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

Initialized empty Git repository in C:/Users/Student/.gitlab-
Student@HNCE-29 MINGW64/ (master)

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) e262bd] 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-program-1.txt

To create/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-program-1.txt onto master branch"

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

\$ git config --global user.name "Pragya-IITGCS05"

\$ git config --global user.email pragya59@gmail.com

(10)
83
10/10

already on 'master'

switched to a new branch 'feature-branch'
Student @ HNNCE-29 MNGW64 ~\Atelphab-1HC21S059 (feature-branch)

[feature branch] 748a5fb] added .gitlab-prgm-2.txt to feature branch
1 file changed, 1 inserted (+)
create mode 100644 .gitlab-prgm-2.txt

Switched to branch 'master'

Student @ HNNCE-29 MNGW64 ~\Atelphab-1HC21S058 (master)

featurebranch -> feature -> feature (Feature-branch)
1 file changed, 1 insertion(+), 1 deletion(-)
diff --git a/featurebranch/a.java
--- a/featurebranch/a.java
+++ b/featurebranch/a.java
@@ -1 +1 @@ -public class FeatureBranch { +public class FeatureBranch {
 public void doSomething() {
 System.out.println("Doing something in Feature Branch");
 }
}

featurebranch -> feature -> feature (Feature-branch)
1 file changed, 1 insertion(+), 1 deletion(-)
diff --git a/featurebranch/a.java
--- a/featurebranch/a.java
+++ b/featurebranch/a.java
@@ -1 +1 @@ -public class FeatureBranch { +public class FeatureBranch {
 public void doSomething() {
 System.out.println("Doing something in Feature Branch");
 }
}

featurebranch -> feature -> feature (Feature-branch)
1 file changed, 1 insertion(+), 1 deletion(-)
diff --git a/featurebranch/a.java
--- a/featurebranch/a.java
+++ b/featurebranch/a.java
@@ -1 +1 @@ -public class FeatureBranch { +public class FeatureBranch {
 public void doSomething() {
 System.out.println("Doing something in Feature Branch");
 }
}

WY featurebranch -> feature -> feature (Feature-branch)
1 file changed, 1 insertion(+), 1 deletion(-)
diff --git a/featurebranch/a.java
--- a/featurebranch/a.java
+++ b/featurebranch/a.java
@@ -1 +1 @@ -public class FeatureBranch { +public class FeatureBranch {
 public void doSomething() {
 System.out.println("Doing something in Feature Branch");
 }
}

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 sure changes in feature branch by adding/modifying/deleting files as needed.

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

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

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

```
$ git commit -m "added gitlab-prog-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 -262bd1..474fa5f

Fast forward

gitlab-program-2.txt | 1 +

1 file changed, 1 insertion(+)

create mode 100644 gitlab-program-2.txt

commit 474fa5f71... (HEAD -> master, feature-branch)

Author: Prajwal-1HC21C5058 <prajwal58@gmail.com>

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

added gitlab-program-2 to feature branch

commit -262bd1031fa

Author: Prajwal-1HC21C5058 <prajwal58@gmail.com>

Date: Thu Oct 30 16:38:10 2025 +0530

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

Updating e262bd1... 474fa5f

Fast forward

gitlab-prgm-2.txt | 1 +

1 file changed, 1 insertion(+)

create mode 100644 gitlab-prgm-2.txt

commit 474fa5f71 ... (HEAD -> master, feature-branch)

Author: Prajwal-1HC21C5058 <prajwal58@gmail.com>

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

added gitlab-prgm-2 to feature branch

commit e262bd1031fa

Author: Prajwal-1HC21C5058 <prajwal58@gmail.com>

Date: Thu Oct 30 16:38:10 2025 +0530

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

Experiment Number Date :

\$ git merge feature-branch

To check logs.

\$ git log

~~10/10~~ 8/10 10/10/20

```
git log --oneline --graph --all --date=iso8601 --abbrev=10 --max-count=1000 --reverse --color=auto
commit: 47450d9747f0080aa204c271e10308043 - 2020-01-20 10:45:00 +0000 - Feature-Branch, feature-branch, merged=1-HC4C80001-merge1@taggit1.com
author: merged=1-HC4C80001-merge1@taggit1.com
date: 2020-01-20 10:45:00 +0000

    add pitch-align-2 to Feature branch

commit: 47450d9747f0080aa204c271e10308043 - 2020-01-20 10:45:00 +0000 - Feature-Branch, feature-branch, merged=1-HC4C80001-merge1@taggit1.com
author: merged=1-HC4C80001-merge1@taggit1.com
date: 2020-01-20 10:45:00 +0000

    add pitch-align-2 to Feature branch

commit: 47450d9747f0080aa204c271e10308043 - 2020-01-20 10:45:00 +0000 - Feature-Branch, feature-branch, merged=1-HC4C80001-merge1@taggit1.com
author: merged=1-HC4C80001-merge1@taggit1.com
date: 2020-01-20 10:45:00 +0000

    add pitch-align-2 to Feature branch

commit: 47450d9747f0080aa204c271e10308043 - 2020-01-20 10:45:00 +0000 - Feature-Branch, feature-branch, merged=1-HC4C80001-merge1@taggit1.com
author: merged=1-HC4C80001-merge1@taggit1.com
date: 2020-01-20 10:45:00 +0000

    add pitch-align-2 to Feature branch

commit: 47450d9747f0080aa204c271e10308043 - 2020-01-20 10:45:00 +0000 - Feature-Branch, feature-branch, merged=1-HC4C80001-merge1@taggit1.com
author: merged=1-HC4C80001-merge1@taggit1.com
date: 2020-01-20 10:45:00 +0000

    add pitch-align-2 to Feature branch

commit: 47450d9747f0080aa204c271e10308043 - 2020-01-20 10:45:00 +0000 - Feature-Branch, feature-branch, merged=1-HC4C80001-merge1@taggit1.com
author: merged=1-HC4C80001-merge1@taggit1.com
date: 2020-01-20 10:45:00 +0000

    add pitch-align-2 to Feature branch
```

No local changes to save

saved working directory and ~~understage~~ on master - stash
temporary storage on master branch

on branch master

changes to be committed:

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

new file: gitlab-prgm-3.txt

gitlab-prgm-3.txt

switched to branch feature-branch

Student@HNNECE-29 MINGW04 ~\gitlab-14C24CS058 (feature-branch)

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 tpt, 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-prgm-3.txt

On branch feature-branch

Changes to be committed:

(use "git restore --staged <file>..." to unstage)
new file: gitlab-prgm-3.txt

[feature-branch 0x9d82bf] add gitlab-prgm-3.txt to feature^{branch}
1 file changed, 1 insertion (+)
create mode 100644 gitlab-prgm-3.txt

commit 0x9d82bf HEAD -> feature-branch

Author: Prajwal-1HCWCS058 <prajwals8@gmail.com>

Date: Thu Oct 30 15:16:01 2025 +0530

add gitlab-prgm-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.tlt` is not committed to git repository, it is not visible in the repository logs. So to commit
\$ git commit -m "add git://prgm-3.tlt to feature branch"

To check logs

\$ git log

~~10/10~~ 87
10/11/15