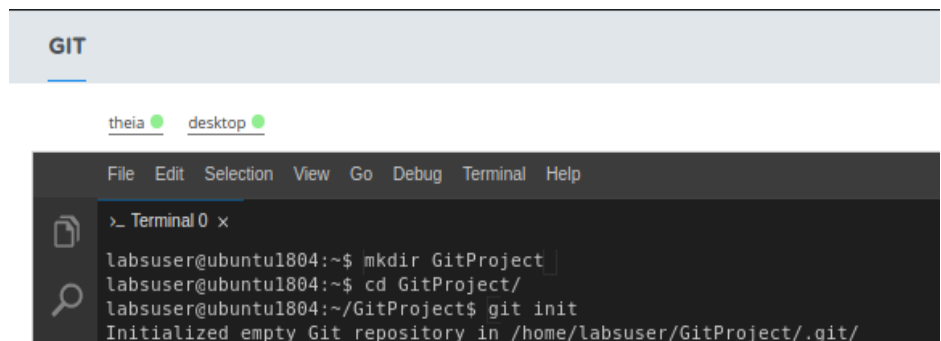


Steps to perform:

1. Start with the Production branch (master branch), and then create a HotFix and Integration branch
2. Subsequently, create Feature 1 and 2 branches that integrate to the Integration branch as shown in the above figure
3. Commit some changes in the Feature 2 branch and merge it into the Integration branch. Delete this branch once merging is complete
4. Commit some changes in the Feature 1 branch and rebase it to the Integration branch
5. Merge the Integration branch into Hotfix and Production branch to update these branches
6. Commit some changes in Feature 1 branch, and then merge it into Integration, Hotfix, and Production branch. Delete this branch once merging is complete
7. **Commit some changes in the Hotfix branch and merge it into the Production as well as the Integration branch**

Steps :

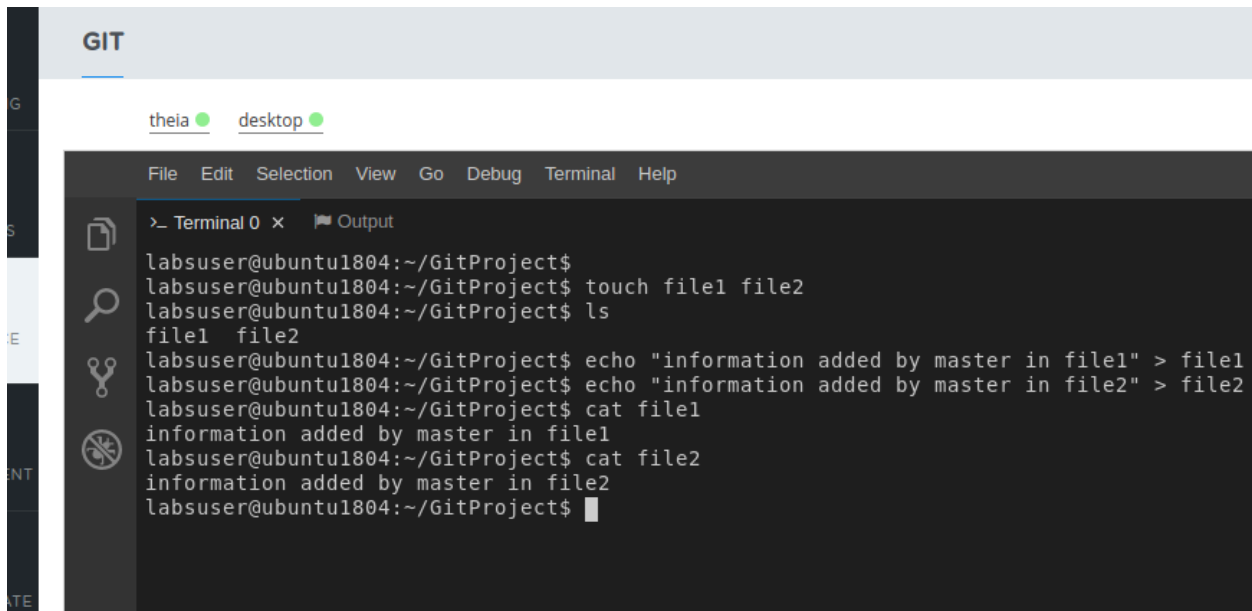
1. Create a directory for the project (in my case i have created GitProject)
mkdir GitProject
Enter in the directory.
cd GitProject
Initialize the git
git init



The screenshot shows a terminal window titled "Terminal 0" within the Theia IDE. The terminal output displays the commands and their results for creating and initializing a Git project:

```
labsuser@ubuntu1804:~$ mkdir GitProject
labsuser@ubuntu1804:~$ cd GitProject/
labsuser@ubuntu1804:~/GitProject$ git init
Initialized empty Git repository in /home/labsuser/GitProject/.git/
```

2. Create two files
touch file1 file2



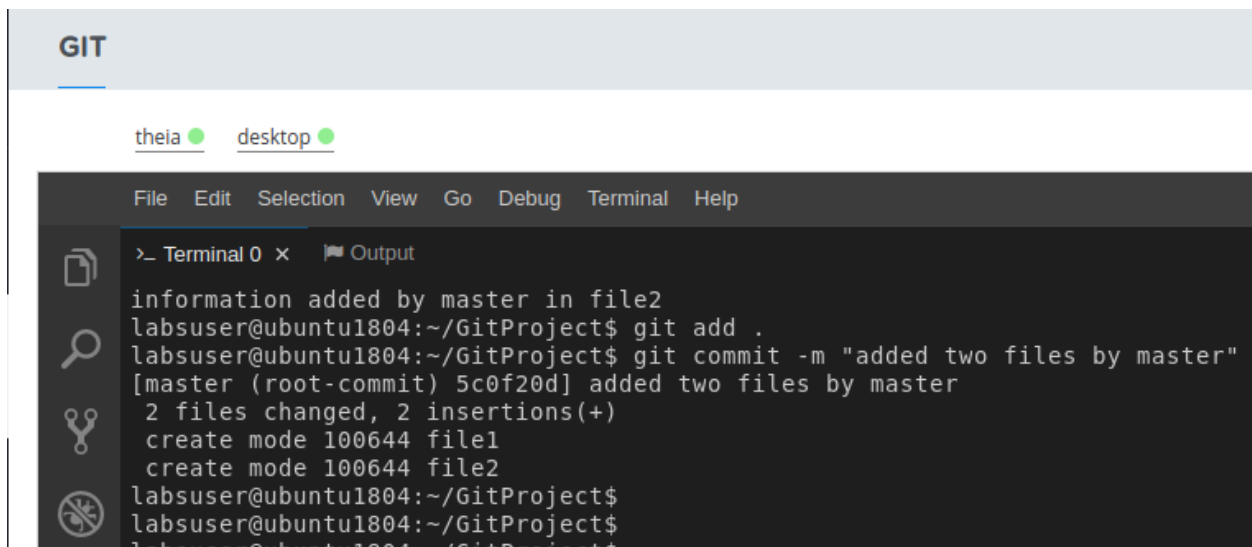
The screenshot shows a terminal window in Theia IDE. The terminal is titled "Terminal 0" and shows the following commands and output:

```
labsuser@ubuntu1804:~/GitProject$ touch file1 file2
labsuser@ubuntu1804:~/GitProject$ ls
file1 file2
labsuser@ubuntu1804:~/GitProject$ echo "information added by master in file1" > file1
labsuser@ubuntu1804:~/GitProject$ echo "information added by master in file2" > file2
labsuser@ubuntu1804:~/GitProject$ cat file1
information added by master in file1
labsuser@ubuntu1804:~/GitProject$ cat file2
information added by master in file2
labsuser@ubuntu1804:~/GitProject$
```

Add some lines to it

echo "information added by master in file 1" > file1
echo "information added by master in file 2" > file2

3. Now we will add these files to the repository
git add .
Commit the changes
git commit -m "added two files by master"



The screenshot shows a terminal window in Theia IDE. The terminal is titled "Terminal 0" and shows the following commands and output:

```
information added by master in file2
labsuser@ubuntu1804:~/GitProject$ git add .
labsuser@ubuntu1804:~/GitProject$ git commit -m "added two files by master"
[master (root-commit) 5c0f20d] added two files by master
2 files changed, 2 insertions(+)
create mode 100644 file1
create mode 100644 file2
labsuser@ubuntu1804:~/GitProject$
labsuser@ubuntu1804:~/GitProject$
labsuser@ubuntu1804:~/GitProject$
```

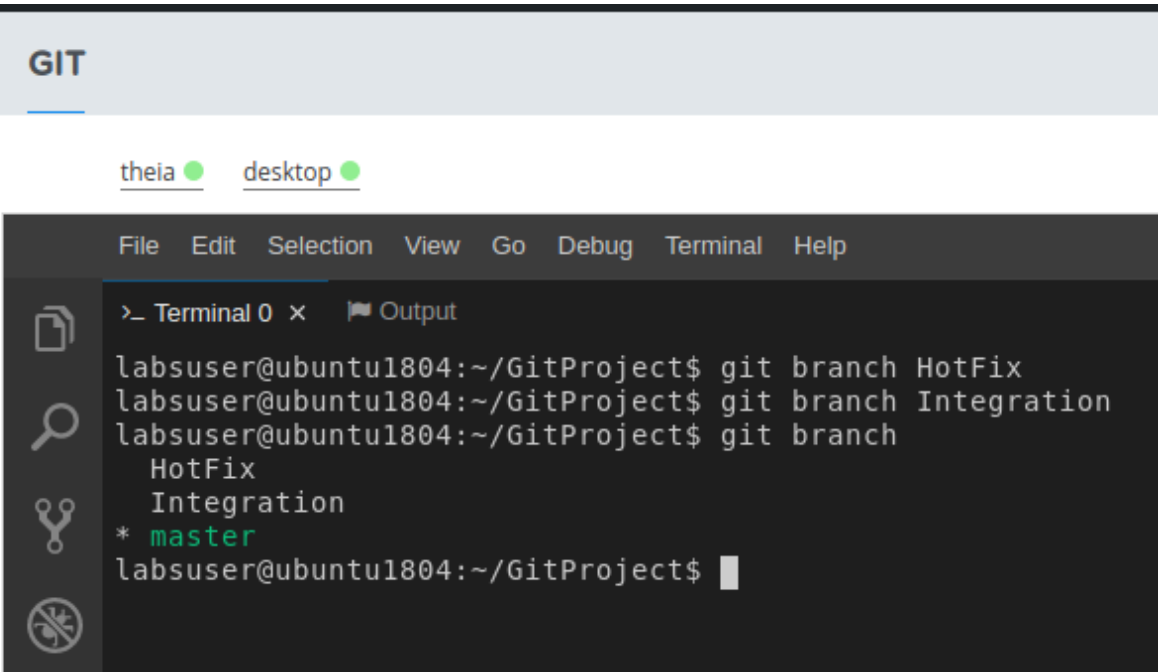
4. Create two branches HotFix and Integration

git branch HotFix

git branch Integration

List the branches and make sure you are on the master branch

git branch



The screenshot shows a terminal window within Theia IDE. The terminal title bar includes 'theia' and 'desktop' with green status indicators. The menu bar contains 'File', 'Edit', 'Selection', 'View', 'Go', 'Debug', 'Terminal', and 'Help'. The terminal content shows the following commands and output:

```
>_ Terminal 0 x Output
labsuser@ubuntu1804:~/GitProject$ git branch HotFix
labsuser@ubuntu1804:~/GitProject$ git branch Integration
labsuser@ubuntu1804:~/GitProject$ git branch
  HotFix
  Integration
* master
labsuser@ubuntu1804:~/GitProject$
```

5. Switch to Integration branch

git checkout Integration

Create two more branches

Feature1 Feature2

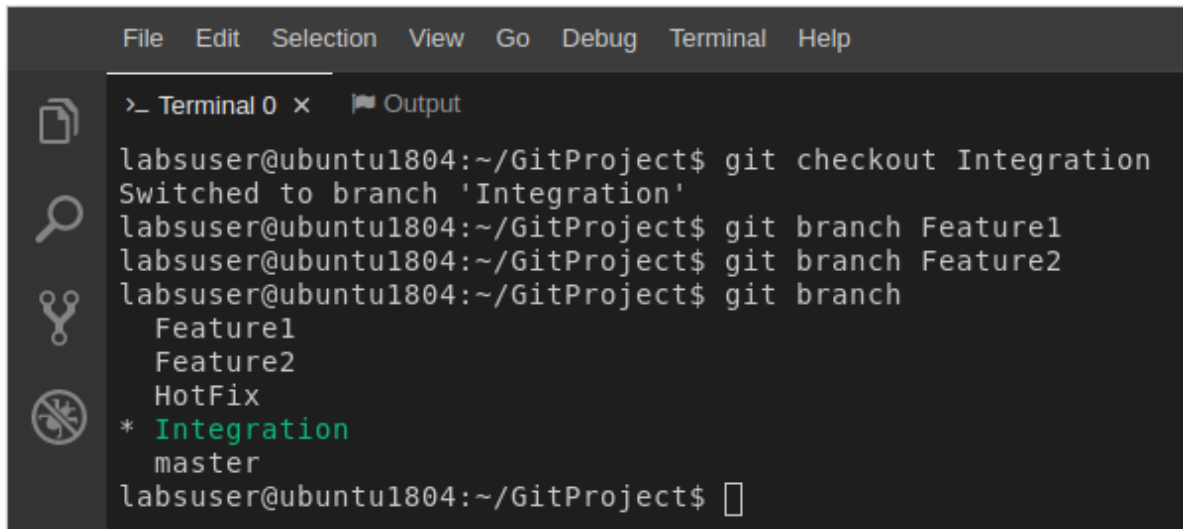
git branch Feature1

git branch Feature2

List the branches again and make sure you are on the Integration branch and Feature1 and Feature2 branches are there.

GIT

theia ● desktop ●



```
File Edit Selection View Go Debug Terminal Help
>_ Terminal 0 x Output
labsuser@ubuntu1804:~/GitProject$ git checkout Integration
Switched to branch 'Integration'
labsuser@ubuntu1804:~/GitProject$ git branch Feature1
labsuser@ubuntu1804:~/GitProject$ git branch Feature2
labsuser@ubuntu1804:~/GitProject$ git branch
  Feature1
  Feature2
  HotFix
* Integration
  master
labsuser@ubuntu1804:~/GitProject$
```

6. Now move to Feature2 branch

git checkout Feature2

Add lines into the files file1 and file2

echo "information added by Feature2 branch in file2" >> file2

echo "information added by Feature2 branch in file1" >> file1

Add and Commit the changes

git commit -a -m "modified two files by Feature2"

GIT

theia  desktop 

```
File Edit Selection View Go Debug Terminal Help

>_ Terminal 0 x Output

labsuser@ubuntu1804:~/GitProject$ git branch
  Feature1
* Feature2
  HotFix
  Integration
  master

labsuser@ubuntu1804:~/GitProject$ echo "information added by Feature2 branch in file2" >> file2
labsuser@ubuntu1804:~/GitProject$ echo "information added by Feature2 branch in file1" >> file1
labsuser@ubuntu1804:~/GitProject$ cat file1
information added by master in file1
information added by Feature2 branch in file1
labsuser@ubuntu1804:~/GitProject$ cat file2
information added by master in file2
information added by Feature2 branch in file2
labsuser@ubuntu1804:~/GitProject$ git commit -a -m "modified two files by Feature2"
[Feature2 5a7b4fb] modified two files by Feature2
 2 files changed, 2 insertions(+)
labsuser@ubuntu1804:~/GitProject$
```

7. Switch back to Integration branch

git checkout Integration

Merge the Feature2 branch with Integration branch

git merge Feature2

Delete the Feature2 branch

git branch -d Feature2

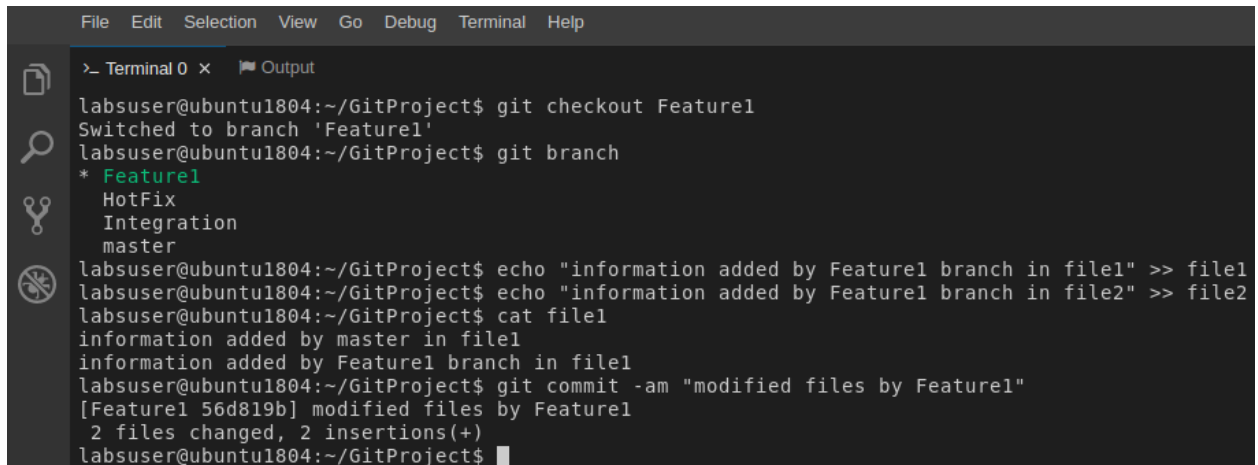
```
File Edit Selection View Go Debug Terminal Help

>_ Terminal 0 x Output

labsuser@ubuntu1804:~/GitProject$ git checkout Integration
Switched to branch 'Integration'
labsuser@ubuntu1804:~/GitProject$ git branch
  Feature1
  Feature2
  HotFix
* Integration
  master

labsuser@ubuntu1804:~/GitProject$ git merge Feature2
Updating 5c0f20d..5a7b4fb
Fast-forward
 file1 | 1 +
 file2 | 1 +
 2 files changed, 2 insertions(+)
labsuser@ubuntu1804:~/GitProject$ git branch -d Feature2
Deleted branch Feature2 (was 5a7b4fb).
```

8. Now switch to Feature1 branch and commit some changes.

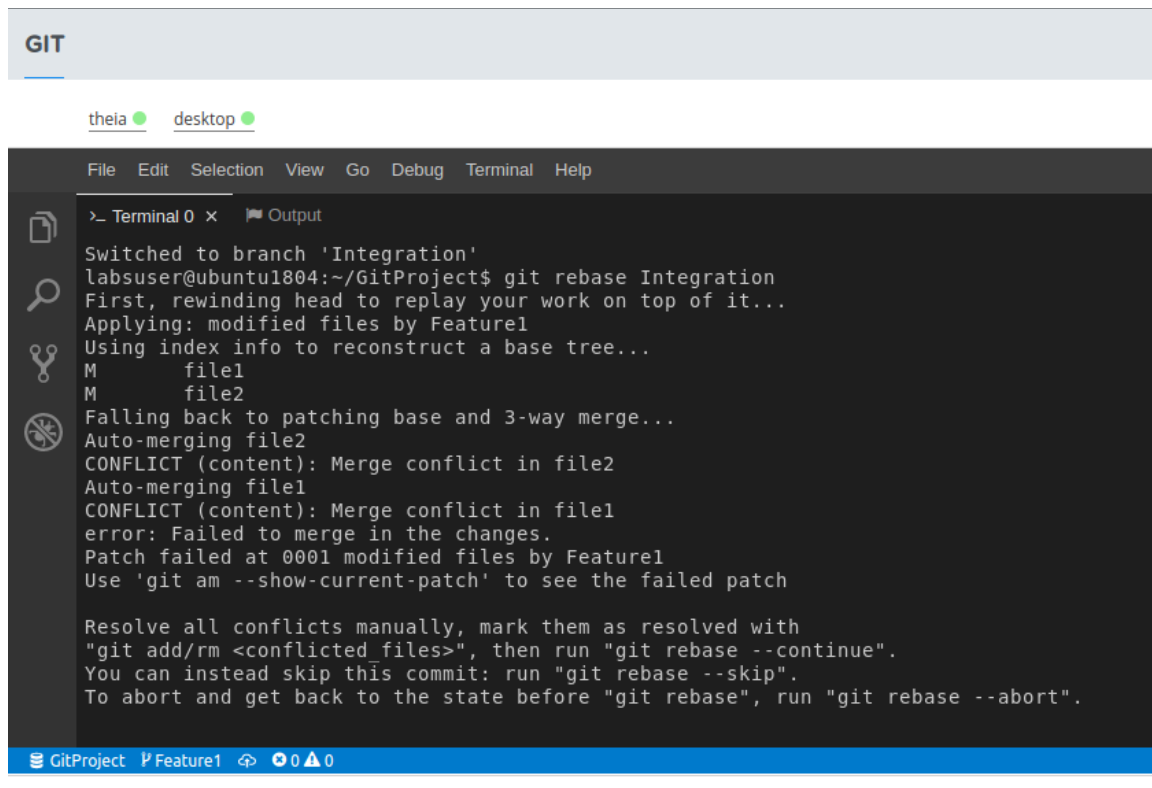


```
File Edit Selection View Go Debug Terminal Help
>_ Terminal 0 x Output
labsuser@ubuntu1804:~/GitProject$ git checkout Feature1
Switched to branch 'Feature1'
labsuser@ubuntu1804:~/GitProject$ git branch
* Feature1
  HotFix
  Integration
  master
labsuser@ubuntu1804:~/GitProject$ echo "information added by Feature1 branch in file1" >> file1
labsuser@ubuntu1804:~/GitProject$ echo "information added by Feature1 branch in file2" >> file2
labsuser@ubuntu1804:~/GitProject$ cat file1
information added by master in file1
information added by Feature1 branch in file1
labsuser@ubuntu1804:~/GitProject$ git commit -am "modified files by Feature1"
[Feature1 56d819b] modified files by Feature1
 2 files changed, 2 insertions(+)
labsuser@ubuntu1804:~/GitProject$
```

9. Now rebase it into the Integration.

git rebase integration

(Git rebase in standard mode will automatically take the commits in your current working branch and apply them to the head of the passed branch)



```

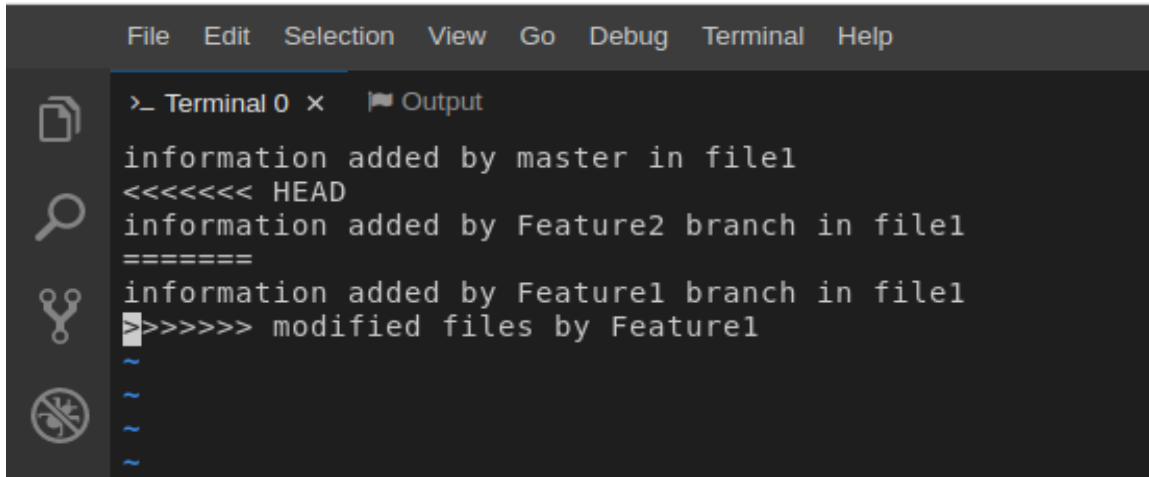
GIT

theia desktop

File Edit Selection View Go Debug Terminal Help
>_ Terminal 0 x Output
Switched to branch 'Integration'
labsuser@ubuntu1804:~/GitProject$ git rebase Integration
First, rewinding head to replay your work on top of it...
Applying: modified files by Feature1
Using index info to reconstruct a base tree...
M   file1
M   file2
Falling back to patching base and 3-way merge...
Auto-merging file2
CONFLICT (content): Merge conflict in file2
Auto-merging file1
CONFLICT (content): Merge conflict in file1
error: Failed to merge in the changes.
Patch failed at 0001 modified files by Feature1
Use 'git am --show-current-patch' to see the failed patch

Resolve all conflicts manually, mark them as resolved with
"git add/rm <conflicted files>", then run "git rebase --continue".
You can instead skip this commit: run "git rebase --skip".
To abort and get back to the state before "git rebase", run "git rebase --abort".

GitProject Feature1 0 0 0
```

A screenshot of a terminal window with a dark background. The terminal shows the output of a git rebase operation. The text is as follows:

```
>_ Terminal 0 x  Output
information added by master in file1
<<<<<<< HEAD
information added by Feature2 branch in file1
=====
information added by Feature1 branch in file1
>>>>>> modified files by Feature1
~
~
~
```

vim file1

or

vim file2

This will show something like the above screenshot. Save and exit.

git add file1 file2

git rebase --continue

A screenshot of a terminal window showing a series of commands and their outputs. The text is as follows:

```
labsuser@ubuntu1804:~/GitProject$ vim file1
labsuser@ubuntu1804:~/GitProject$ vim file2
labsuser@ubuntu1804:~/GitProject$
labsuser@ubuntu1804:~/GitProject$
labsuser@ubuntu1804:~/GitProject$
labsuser@ubuntu1804:~/GitProject$
labsuser@ubuntu1804:~/GitProject$ git add file1 file2
labsuser@ubuntu1804:~/GitProject$ git rebase --continue
Applying: modified files by Feature1
labsuser@ubuntu1804:~/GitProject$
```

10. Merge the Integration branch into Hotfix and Production branch to update these branches

git checkout HotFix

git merge Integration

git checkout Production

git merge integration


```
File Edit Selection View Go Debug Terminal Help
>_ Terminal 0 x Output
labsuser@ubuntu1804:~/GitProject$ git checkout HotFix
Switched to branch 'HotFix'
labsuser@ubuntu1804:~/GitProject$ git merge Integration
Updating 5c0f20d..5a7b4fb
Fast-forward
 file1 | 1 +
 file2 | 1 +
 2 files changed, 2 insertions(+)
labsuser@ubuntu1804:~/GitProject$ git checkout master
Switched to branch 'master'
labsuser@ubuntu1804:~/GitProject$ git merge Integration
Updating 5c0f20d..5a7b4fb
Fast-forward
 file1 | 1 +
 file2 | 1 +
 2 files changed, 2 insertions(+)
```

11. Commit some changes in Feature1 branch, and then merge it into Integration, Hotfix, and Production(i.e master) branch. Delete this branch once merging is complete

```
labsuser@ubuntu1804:~/GitProject$ git checkout Feature1
Switched to branch 'Feature1'
labsuser@ubuntu1804:~/GitProject$ git branch
* Feature1
  HotFix
  Integration
  master
labsuser@ubuntu1804:~/GitProject$ echo "brand new line added by Feature1 in file1" >> file1
labsuser@ubuntu1804:~/GitProject$ echo "brand new line added by Feature1 in file2" >> file2
```

```
labsuser@ubuntu1804:~/GitProject$ git commit -am "new modification by Feature1"
[Feature1 88b76e1] new modification by Feature1
 2 files changed, 2 insertions(+)
labsuser@ubuntu1804:~/GitProject$
```

```
File Edit Selection View Go Debug Terminal Help
>_ Terminal 0 x Output
=====
labsuser@ubuntu1804:~/GitProject$ git branch
* Feature1
  HotFix
  Integration
  master
labsuser@ubuntu1804:~/GitProject$ git checkout Feature1
Switched to branch 'Feature1'
labsuser@ubuntu1804:~/GitProject$ git branch
* Feature1
  HotFix
labsuser@ubuntu1804:~/GitProject$ git checkout Integration
Switched to branch 'Integration'
labsuser@ubuntu1804:~/GitProject$ git merge Feature1
Updating 5a7b4fb..88b76e1
Fast-forward
 file1 | 5 +++++
 file2 | 5 +++++
 2 files changed, 10 insertions(+)
labsuser@ubuntu1804:~/GitProject$ git checkout HotFix
Switched to branch 'HotFix'
labsuser@ubuntu1804:~/GitProject$ git merge Integration
Updating 5a7b4fb..88b76e1
Fast-forward
 file1 | 5 +++++
 file2 | 5 +++++
 2 files changed, 10 insertions(+)
labsuser@ubuntu1804:~/GitProject$ git checkout master
Switched to branch 'master'
labsuser@ubuntu1804:~/GitProject$ git merge Feature1
Updating 5a7b4fb..88b76e1
Fast-forward
 file1 | 5 +++++
 file2 | 5 +++++
 2 files changed, 10 insertions(+)
labsuser@ubuntu1804:~/GitProject$
```

12. Delete the Feature1 branch after this.
git branch -d Feature1

```
labsuser@ubuntu1804:~/GitProject$ git branch -d Feature1
Deleted branch Feature1 (was 88b76e1).
labsuser@ubuntu1804:~/GitProject$
```

13. Commit some changes in the Hotfix branch and merge it into the Production as well as the Integration branch

```
File Edit Selection View Go Debug Terminal Help

>_ Terminal 0 x Output

Fast-forward
 file1 | 5 +++++
 file2 | 5 +++++
2 files changed, 10 insertions(+)
labsuser@ubuntu1804:~/GitProject$ git checkout HotFix
Switched to branch 'HotFix'
labsuser@ubuntu1804:~/GitProject$ git merge Integration
Updating 5a7b4fb..88b76e1
Fast-forward
labsuser@ubuntu1804:~/GitProject$ git checkout HotFix
Switched to branch 'HotFix'
labsuser@ubuntu1804:~/GitProject$ echo "modifications by HotFix branch in File1" >> file1
labsuser@ubuntu1804:~/GitProject$ echo "modifications by HotFix branch in File2" >> file2
labsuser@ubuntu1804:~/GitProject$ cat file1
information added by master in file1
<<<<<<< HEAD
information added by Feature2 branch in file1
=====
information added by Feature1 branch in file1
>>>>>>> modified files by Feature1
brand new line added by Feature1 in file1
modifications by HotFix branch in File1
labsuser@ubuntu1804:~/GitProject$ git commit -am "changes by HotFix branch"
[HotFix 84cf287] changes by HotFix branch
2 files changed, 2 insertions(+)
labsuser@ubuntu1804:~/GitProject$ cat file1
information added by master in file1
<<<<<<< HEAD
information added by Feature2 branch in file1
=====
information added by Feature1 branch in file1
>>>>>>> modified files by Feature1
brand new line added by Feature1 in file1
modifications by HotFix branch in File1
labsuser@ubuntu1804:~/GitProject$ vim file1
labsuser@ubuntu1804:~/GitProject$
```

GIT

[thela](#) [desktop](#)

```
File Edit Selection View Go Debug Terminal Help

>_ Terminal 0 x Output

labsuser@ubuntu1804:~/GitProject$ git checkout Integration
Switched to branch 'Integration'
labsuser@ubuntu1804:~/GitProject$ git merge Hotfix
merge: Hotfix - not something we can merge
labsuser@ubuntu1804:~/GitProject$ git merge HotFix
Updating 88b76e1..84cf287
Fast-forward
 file1 | 1 +
 file2 | 1 +
2 files changed, 2 insertions(+)
labsuser@ubuntu1804:~/GitProject$ git checkout master
Switched to branch 'master'
labsuser@ubuntu1804:~/GitProject$ git merge HotFix
Updating 88b76e1..84cf287
Fast-forward
 file1 | 1 +
 file2 | 1 +
2 files changed, 2 insertions(+)
labsuser@ubuntu1804:~/GitProject$
```