

Subject Name: Source Code Management

Subject Code: CS181

Cluster: Beta

Department: DCSE



Submitted By:

Aashima Mahajan

2110990021

G01

Submitted To:

Dr Monit

Kapoor

List of Programs

S. No	Program Title	Page No.
1.	Add collaborators on Github Repo	1
2.	Fork and Commit	5
3.	Merge and Resolve Conflicts	9
4.	Reset and Revert	12



Experiment No. 06

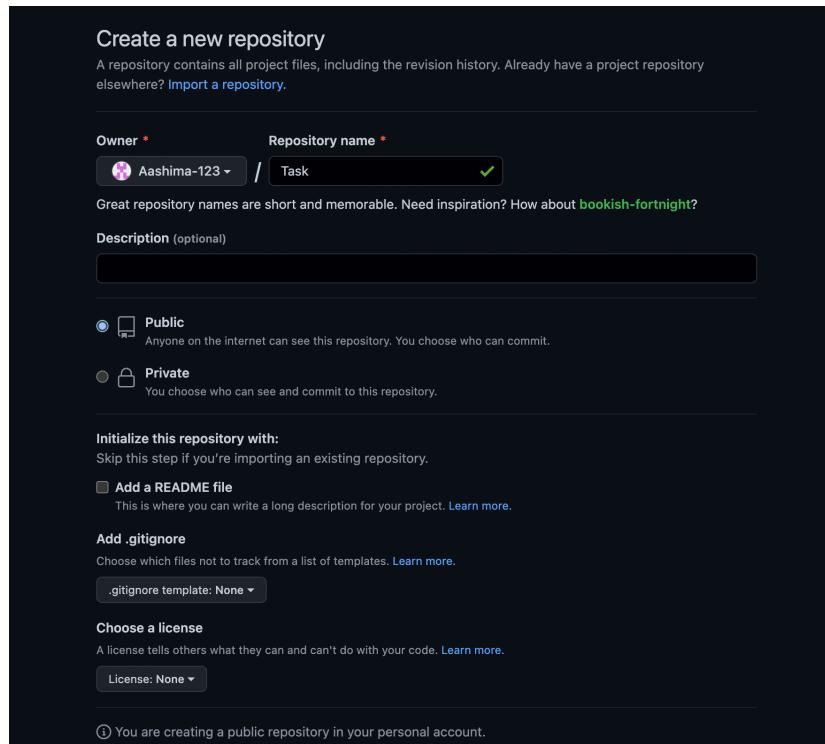
Aim: Add collaborators on Github Repo

Theory:

1. Create a New Repository.

A repository is like a folder for your project. You can have any number of public and private repositories in your user account. Repositories can contain folders and files, images, videos, spreadsheets, and data sets, as well as the revision history for all files in the repository. For more information, see "[About repositories](#)."

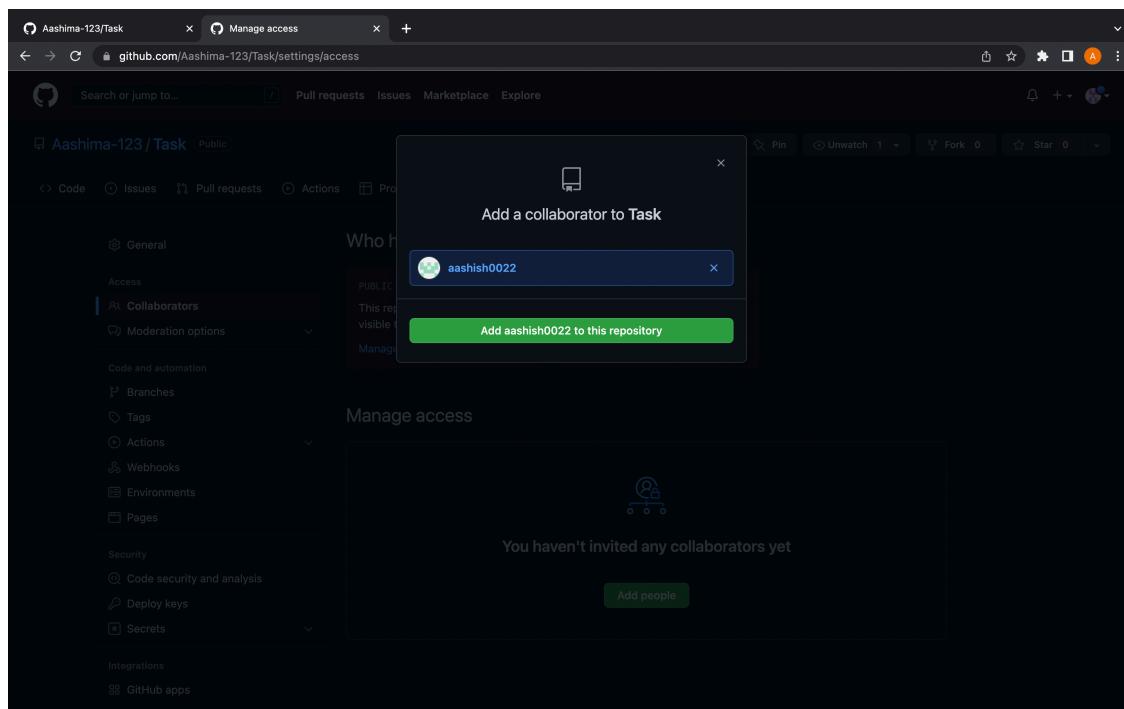
When you create a new repository, you should initialise the repository with a README file to let people know about your project. For more information, see "[Creating a new repository](#)."



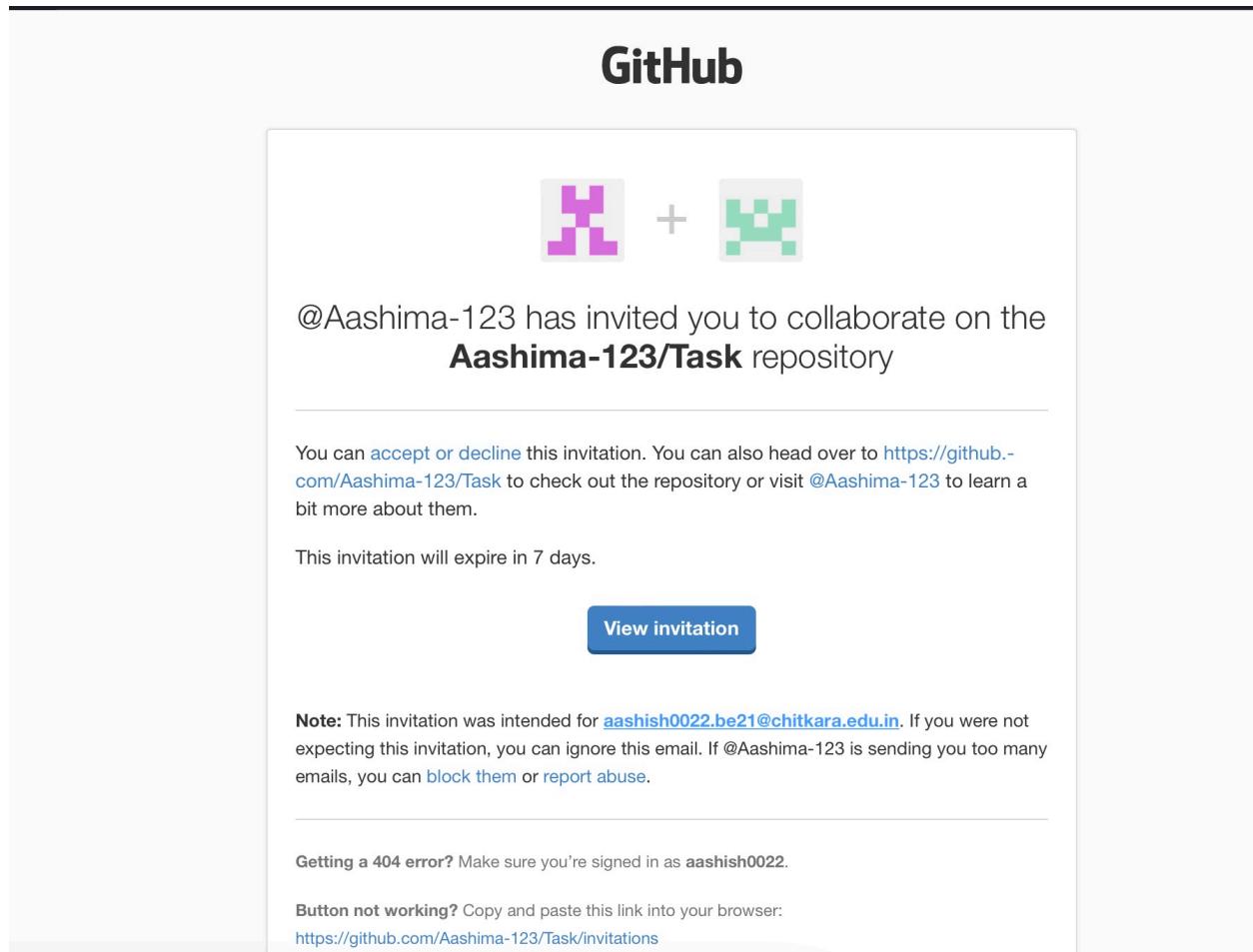
2. Now Copy the HTTP link of your repo and paste it on your 'Git CLI', and merge the local repo in remote repo .

```
Last login: Wed Jun 1 11:24:51 on ttys000
(base) aashima@aashimas-MacBook-Air Task % ls
function.cpp
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /Users/aashima/Desktop/Task/.git/
(base) aashima@aashimas-MacBook-Air Task % git add .
(base) aashima@aashimas-MacBook-Air Task % git commit
hint: Waiting for your editor to close the file... error: There was a problem with the editor 'vi'.
Please supply the message using either -m or -F option.
(base) aashima@aashimas-MacBook-Air Task % git commit -m "Initial commit"
[master (root-commit) a9944bd] Initial commit
 1 file changed, 1 insertion(+)
 create mode 100644 function.cpp
(base) aashima@aashimas-MacBook-Air Task % git remote add origin https://github.com/Aashima-123/Task.git
(base) aashima@aashimas-MacBook-Air Task % git push origin master
Enumerating objects: 3, done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 836 bytes | 836.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Aashima-123/Task.git
 * [new branch]    master -> master
(base) aashima@aashimas-MacBook-Air Task %
```

3. Go to Collaborators in Repo Setting, Add the username or email of Collaborator you want to add in your Repo.



4. Invitation Mail is sent to the Collaborator, the collaborator has to accept this Invitation.



5. New Collaborator has now access to the Aashima-123 Repo.

Searched for "Aashima-123 / Task" (Public)

Pull requests Issues Marketplace Explore

Pin Unwatch 1 Fork 0 Star 0

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Who has access

Access

Collaborators (selected)

Moderation options

Code and automation

Branches Tags Actions Webhooks Environments Pages

Security

Code security and analysis Deploy keys Secrets

Integrations

GitHub apps Email notifications

DIRECT ACCESS

1 has access to this repository. 1 collaborator.

Manage

Manage access

Add people

Select all Type ▾

Find a collaborator...

aashish0022 Collaborator Remove

< Previous Next >

This screenshot shows the 'Settings' page for a public GitHub repository named 'Aashima-123 / Task'. The 'Access' section is selected, specifically the 'Collaborators' tab. It displays that there is 1 collaborator, 'aashish0022', listed under 'DIRECT ACCESS'. The user can manage this collaborator or add more. Below this, the 'Manage access' section allows for selecting all users or adding new ones. The sidebar on the left lists various repository settings like Code and automation, Security, and Integrations.

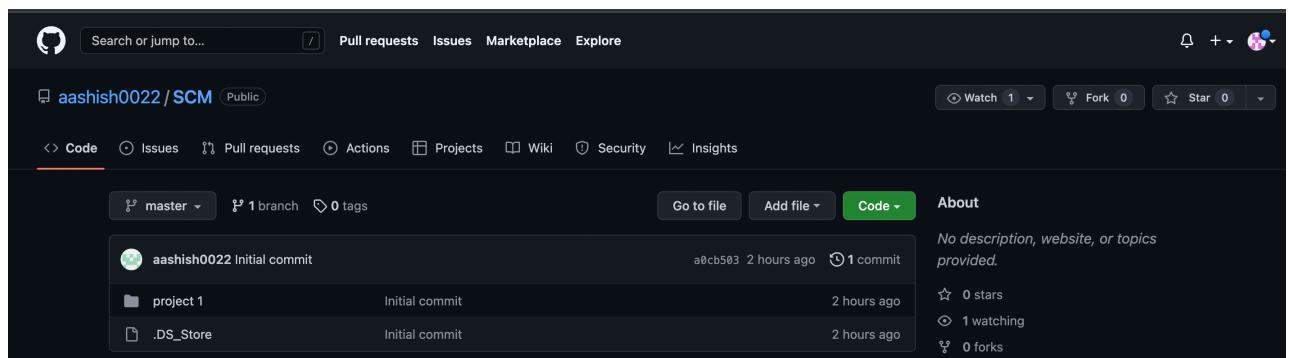
Experiment No. 07

Aim: Fork and Commit

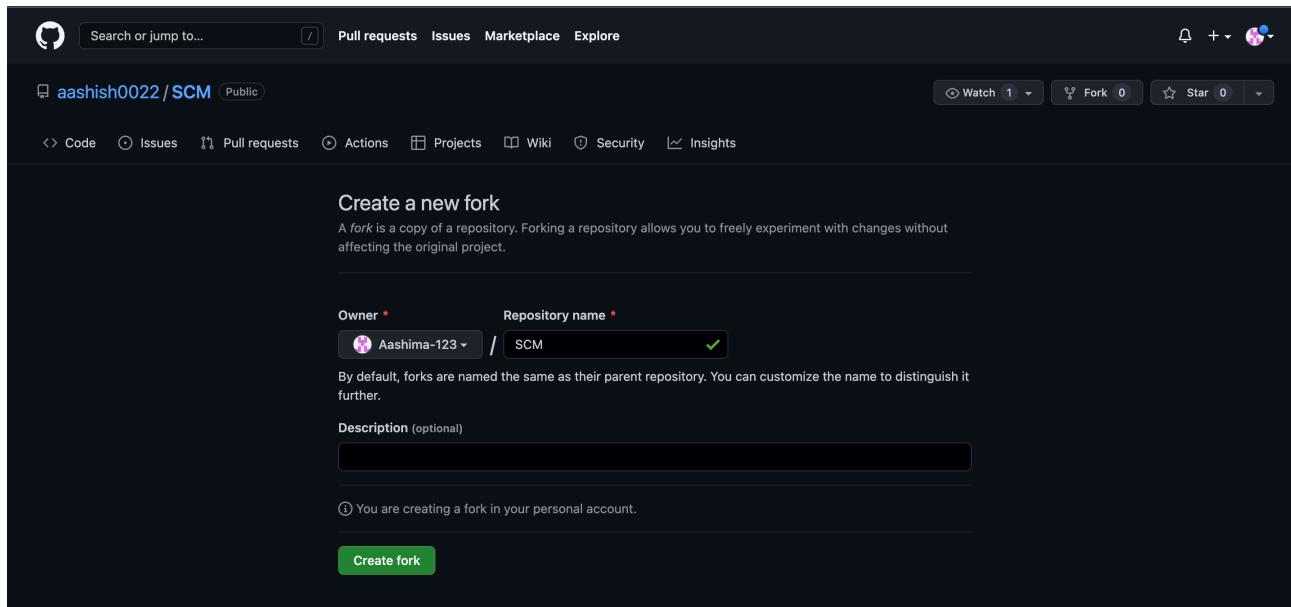
Theory:

Fork :- A Fork is copy of a repository . Forking a repository allows you to freely experiment with changes without affecting the original Project.

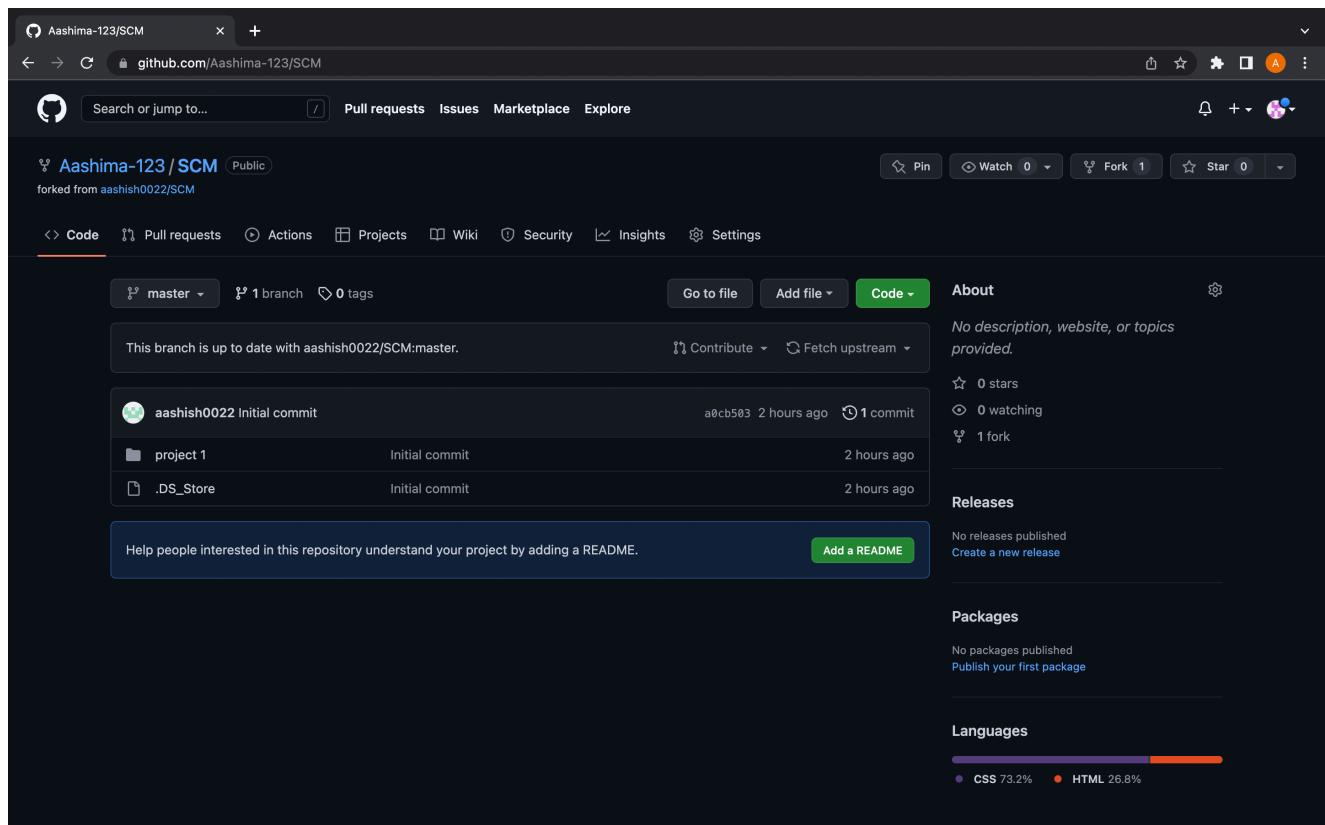
1. To fork a repository first thing you need to do is, sign in to your GitHub account and then you came to the repository you want to fork, so here for demo purpose am using aashish0022 repository.



2. Click on the **Fork** button on right upside corner. Then it will ask to create a new fork, add description if you want and then click on create fork.



3. Now you will have a copy of the repo you have forked from other user. Now you can do any modification you want without making changes to main source code.



4. Now type <https://github.com/Aashima-123/SCM> on CLI.

Git pull <url> --> This command is used to fetch the remote repo or to clone the repo.

```
Last login: Wed Jun 1 20:32:45 on ttys000
(base) aashima@aashimas-MacBook-Air scm % git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /Users/aashima/Desktop/scm/.git/
(base) aashima@aashimas-MacBook-Air scm % git clone https://github.com/Aashima-123/SCM
Cloning into 'SCM'...
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 7 (delta 1), reused 7 (delta 1), pack-reused 0
Receiving objects: 100% (7/7), done.
Resolving deltas: 100% (1/1), done.
(base) aashima@aashimas-MacBook-Air scm % ls
2110990021 SCM.pages    2110990021 SCM.pdf    2110990021.pages    2110990021.pdf    SCM

```

5. Now Open the file make changes in it and commit it and push it to remote.

```
(base) aashima@aashimas-MacBook-Air SCM % nano
(base) aashima@aashimas-MacBook-Air SCM % git add .
(base) aashima@aashimas-MacBook-Air SCM % git commit -m"added changes"
[master fa94974] added changes
 1 file changed, 1 insertion(+)
 create mode 100644 changes
(base) aashima@aashimas-MacBook-Air SCM % git push origin master
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 331 bytes | 331.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Aashima-123/SCM
 a0cb503..fa94974 master -> master
(base) aashima@aashimas-MacBook-Air SCM %
```

6. Now go to GitHub and accept the merge request.

Search or jump to... Pull requests Issues Marketplace Explore

Watch 1 Fork 1 Star 0

aashish0022 / SCM Public

Code Issues Pull requests Actions Projects Wiki Security Insights

added changes #1

Open Aashima-123 wants to merge 1 commit into `aashish0022:master` from `Aashima-123:master`

Conversation 0 Commits 1 Checks 0 Files changed 1 +1 -0

Aashima-123 commented now
No description provided.

added changes fa94974

Add more commits by pushing to the `master` branch on **Aashima-123/SCM**.

This branch has no conflicts with the base branch
Only those with [write access](#) to this repository can merge pull requests.

Write Preview

Leave a comment

Reviewers
No reviews
Still in progress? Convert to draft

Assignees
No one assigned

Labels
None yet

Projects
None yet

Milestone
No milestone

Development

Experiment No. 08

Aim: Merge and Resolve Conflicts

Theory:

1. Do changes in master branch and commit those change. And checkout to different branch and again do changes and commit it. Now checkout to master branch and merge that branch in master.

COMMIT IN MASTER BRANCH

COMMIT IN ACTIVITY BRANCH

2. Now try to merge it will give Conflicts Error.

3. Use Command “git mergetool” to solve the conflict.

git -mergetool – Run merge conflict resolution tools to resolve merge conflicts.

4. Press “I” to insert, after insertion . Press “:wq”. The merge conflict is solved and our Activity branch is merged to master branch.

```
MINGW64 /c/Users/jakob/Desktop/SCM1
```

The screenshot shows a terminal window with three tabs. The tabs are labeled 'hello friends', 'hello', and 'file1'. The 'hello friends' tab is active and displays the following command-line session:

```
./file1.LOCAL_482.txt [dos] (15:12 02/05/2022) 1,3-1 All ./file1.BASE_482.txt [dos] (15:12 02/05/2022) 1,5-1 All ./file1.REMOTE_482.txt [dos] (15:12 02/05/2022) 2,5-1 All
[executing]
cccccc HEAD
aaaaaa
bnn1o
oooooo activity
```


Experiment No. 09

Aim: Reset and Revert

Theory:

Git-revert – Revert some existing commits.

1. On GitBash CLI, Type command “git revert <Commit id>”. It revert the changes that done before Commit.

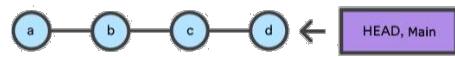
```
(base) aashima@aashimas-MacBook-Air:2416090884 % cd ..
(base) aashima@aashimas-MacBook-Air:Desktop % mkdir sm
(base) aashima@aashimas-MacBook-Air:Desktop % cd sm
(base) aashima@aashimas-MacBook-Air:sm % touch first.cpp
(base) aashima@aashimas-MacBook-Air:sm % vi first.cpp
(base) aashima@aashimas-MacBook-Air:sm % git init
hint: Using `master` as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
(base) aashima@aashimas-MacBook-Air:sm % git branch -m names
Initialized empty Git repository in /Users/aashima/Desktop/sm/.git/
(base) aashima@aashimas-MacBook-Air:sm % git add .
(base) aashima@aashimas-MacBook-Air:sm % git commit -m "Added a comment"
[master (root-commit) bf340cf] Added a comment
1 file changed, 1 insertion(+)
create mode 100644 first.cpp
(base) aashima@aashimas-MacBook-Air:sm % vi first.cpp
(base) aashima@aashimas-MacBook-Air:sm % git add .
(base) aashima@aashimas-MacBook-Air:sm % git commit -m "Second Commit"
[master 93f29c2] Second Commit
1 file changed, 1 insertion(+)
(base) aashima@aashimas-MacBook-Air:sm % git log --oneline
93f29c2 [HEAD -> master] Second Commit
bf340cf Added a comment
(base) aashima@aashimas-MacBook-Air:sm % cat first.cpp
// added a comment
// added another comment
(base) aashima@aashimas-MacBook-Air:sm % git revert bf340cf
CONFLICT (modify/delete): first.cpp deleted in (empty tree) and modified in HEAD. Version HEAD of first.cpp left in tree.
error: could not revert bf340cf... Added a comment
hint: conflict while switching to a different branch. Make sure you corrected paths
hint: with 'git add <paths>' or 'git rm <paths>'
hint: and commit the result with 'git commit'
(base) aashima@aashimas-MacBook-Air:sm % git revert --abort
(base) aashima@aashimas-MacBook-Air:sm % git revert 93f29c2
[master a1fcf81] Revert "Testing Revert"
1 file changed, 1 deletion(-)
(base) aashima@aashimas-MacBook-Air:sm % git log --oneline
a1fcf81 [HEAD -> master] Revert "Testing Revert"
93f29c2 Second Commit
bf340cf Added a comment
(base) aashima@aashimas-MacBook-Air:sm % cat first.cpp
// added a comment
(base) aashima@aashimas-MacBook-Air:sm %
```

git revert HEAD~3 :-

Revert the changes specified by the fourth last commit in HEAD and create a new commit with the reverted changes

Git-reset - Reset current HEAD to the specified state

2. At a surface level, git reset is similar in behavior to git checkout. Where git checkout solely operates on the HEAD ref pointer, git reset will move the HEAD ref pointer and the current branch ref pointer. To better demonstrate this behaviour consider the following example. This example demonstrates a sequence of commits on the main branch. The HEAD ref and main branch ref currently point to commit d. Now let us execute and compare, both git checkout b and git reset b.



GIT RESET:

1. **reset** is the command we use when we want to move the repository back to a previous **commit**, discarding any changes made after that **commit**.

```
[(base) aashima@aashimas-MacBook-Air sm % vi first.cpp
[(base) aashima@aashimas-MacBook-Air sm % git add .
[(base) aashima@aashimas-MacBook-Air sm % git commit -m "Added some comments"
[master 6fa81ee] Added some comments
 1 file changed, 2 insertions(+)
[(base) aashima@aashimas-MacBook-Air sm % git log --oneline
6fa81ee (HEAD -> master) Added some comments
a11fc81 Revert "Testing Revert"
93f29c2 Second Commit
bf340cf Added a comment
[(base) aashima@aashimas-MacBook-Air sm % git reset --hard a11fc81
HEAD is now at a11fc81 Revert "Testing Revert"
[(base) aashima@aashimas-MacBook-Air sm % git log --oneline
a11fc81 (HEAD -> master) Revert "Testing Revert"
93f29c2 Second Commit
bf340cf Added a comment
(base) aashima@aashimas-MacBook-Air sm % ]
```