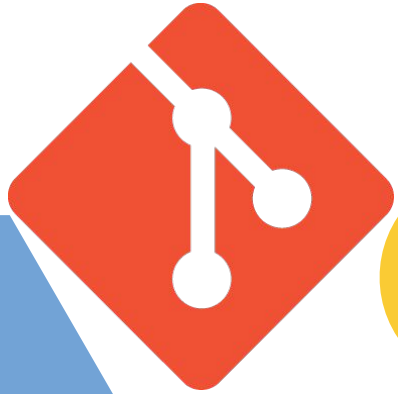
Version Control using Git



Coding

- All the design and thinking get transformed to an implementation.
- The main outcome is the code
- Software Engineer Vs Programmer

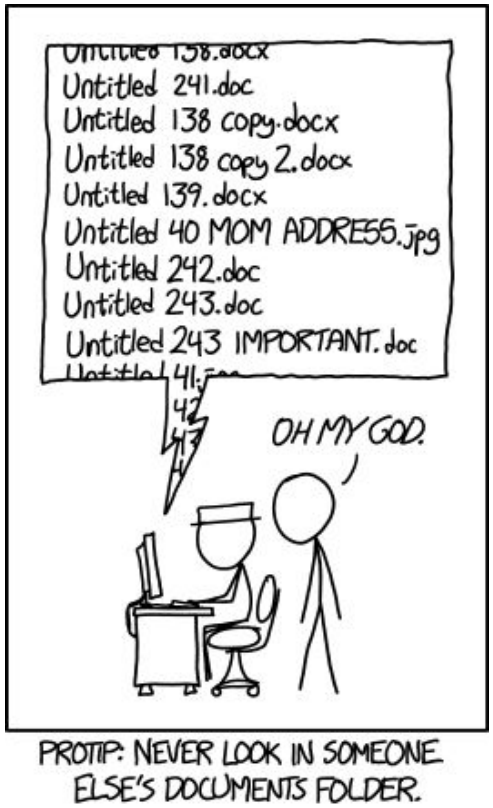
Need for Version Control

```
void drawPicture() {
  drawCircle();
  for (int i = 0; i < 10; i++) {
    drawTriangle();
  }
}
```

```
void drawPicture() {
  drawCircle();
  for (int i = 0; i < 10; i++) {
    drawTriangle();
  }
}
```

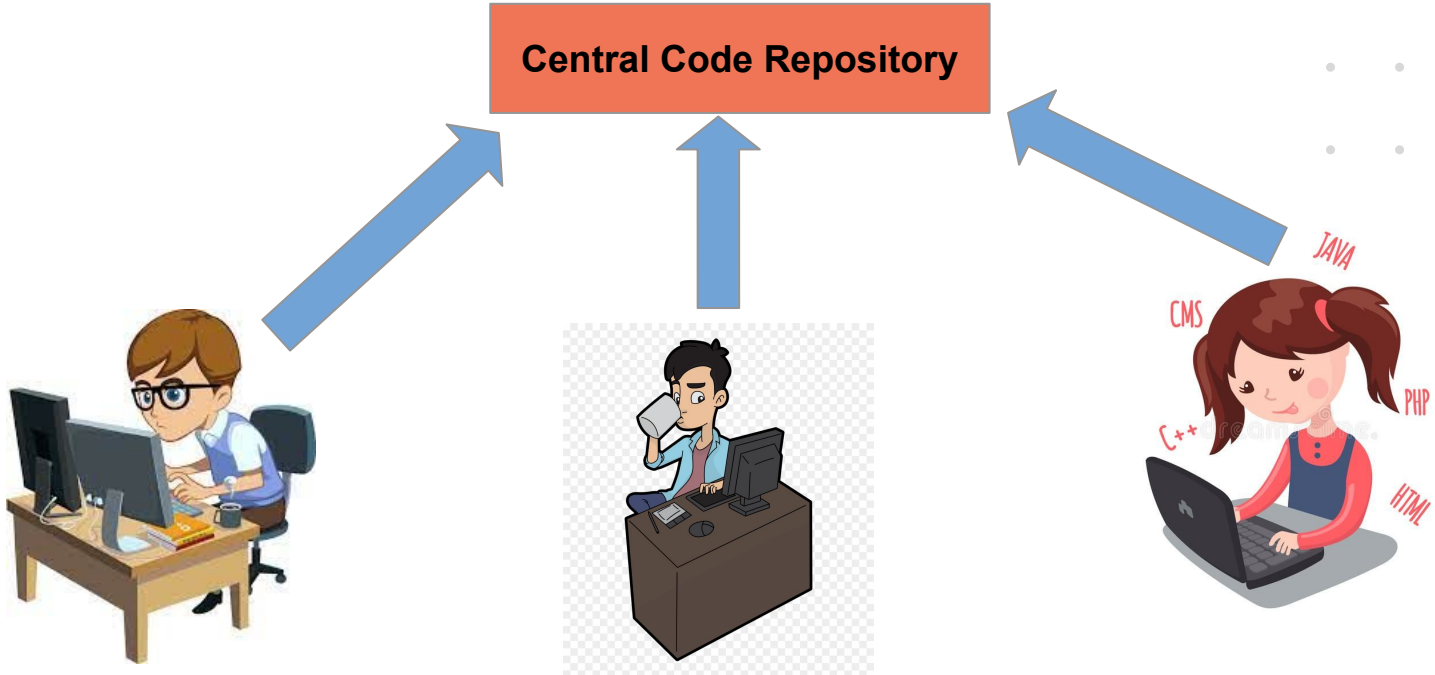
```
void drawPicture() {
  drawCircle();
  for (int i = 0; i < 100; i++)
{
    drawSquare();
  }
}
```

Need for Version Control



Source: https://geo-python-site.readthedocs.io/en/latest/lessons/L2/intro-to-GitHub.html

Need for Version Control



Git

- Most widely used version control system
- Developed by Linus Torvalds in 2005
- Git is an example of Distributed Version Control System (DVCS)

Setting up a new Repository

To create a new repository for your new project

Project path: `/home/user/projects/sample_proj`

```
cd /home/user/projects/sample_proj
git init
```

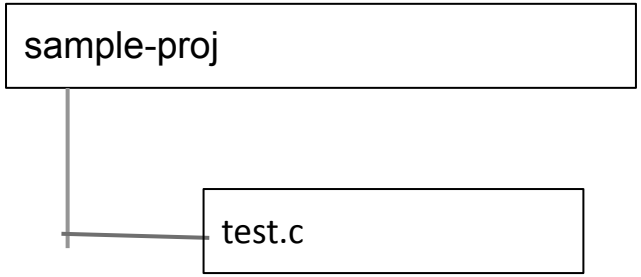
This will initialize git and a `.git` folder will get created inside the project path

Setting up a new Repository: Config

Config is to get or set global properties for the repository

```
git config --global user.email "you@example.com"
git config --global user.name "Your Name"
```

Setting up a new Repository



Git status

To know the state of the files

```
git status
```

```
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        test.c

nothing added to commit but untracked files present (use "git add" to track)
```

States of a file

Untracked: When a new file is created it is untracked

Staged: A staged file is ready to go into the local database during the next commit

Unmodified/Commited: the file is safely stored in the local git database.

Modified: file is has changes which are yet to be committed

Existing Repository

```
git clone <repo url>
```

Git supports a few different network protocols (http, ssh etc.)

```
git clone https://github.com/linux-test-project/ltp.git
```

After cloning, all the files in the repository are in unmodified/committed state.

```
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
```

Staging the files

Staged files get committed in the next commit

```
git add
```

```
On branch master

No commits yet

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

How to Commit?

- When a file is committed, it becomes a part of the local repository.
- All the staged files get committed

`git commit -m <commit message>`

```
git commit -m “first commit”
```

```
[master (root-commit) 44329c7] first commit
 1 file changed, 4 insertions(+)
 create mode 100644 test.c
```

```
git status
```

```
On branch master
nothing to commit, working tree clean
```

Source: https://dev.to/sublimegeek/git-staging-area-explained-like-im-five-1anh

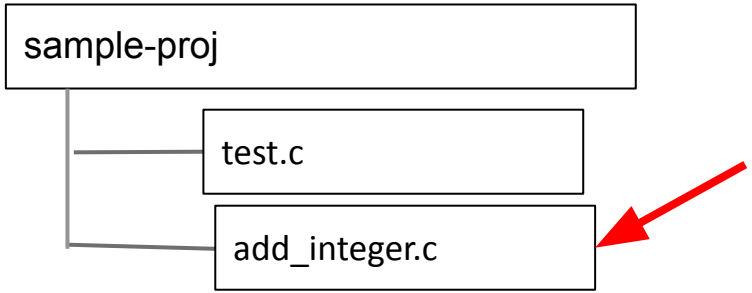
Viewing commit history

```
git log
```

Shows commits made in the repository in the reverse chronological order.

```
commit 44329c79f226226bca9507b90d9ddfc5217f2fe2 (HEAD -> master)
Author: Joe Cheri Ross <joecheri@gmail.com>
Date:   Fri Dec 30 16:51:24 2022 +0530

    first commit
```

More on Commit



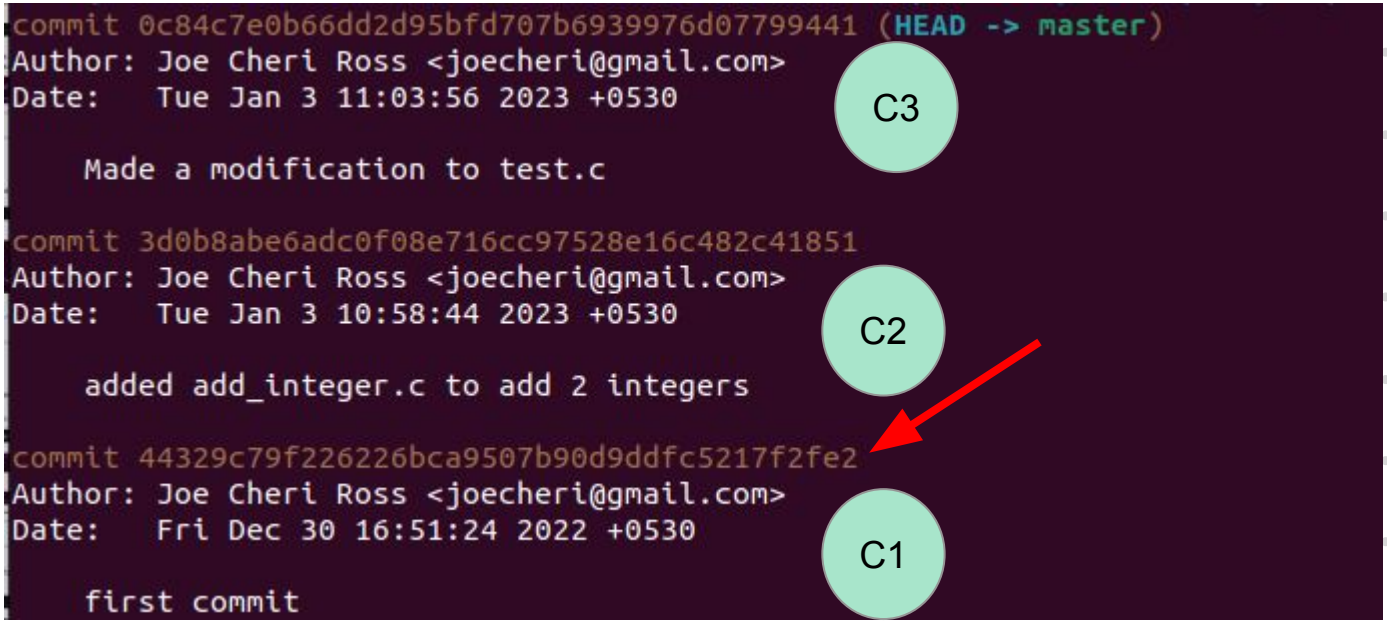

```
git add add_integer.c
git commit -m “added add_integer.c to add 2 integers”
```

Modified test.c

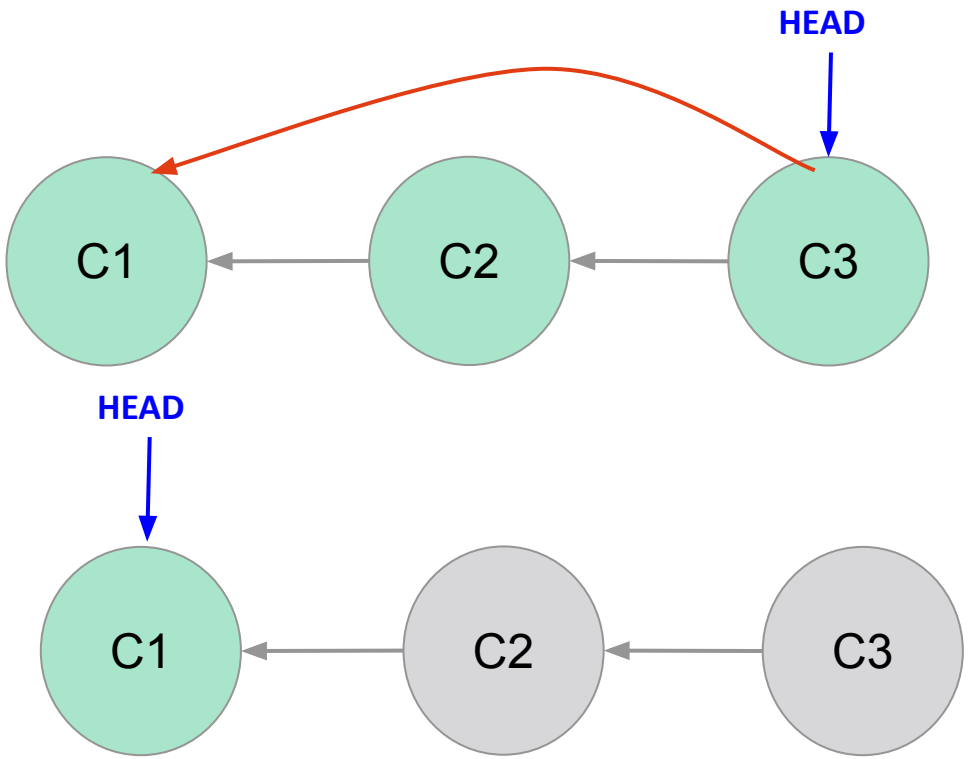
```
git add test.c
git commit -m “Made a modification to test.c”
```

More on Commit

```
git log
```



Undoing things



Undoing Things: go back to a previous commit

`git reset <commit id>`

Reverts to the previous commit referred by the commit id

```
git reset 44329c79f226226bca9507b90d9ddfc5217f2fe2
```

```
git log
```

```
commit 44329c79f226226bca9507b90d9ddfc5217f2fe2 (HEAD -> master)
Author: Joe Cheri Ross <joecheri@gmail.com>
Date:   Fri Dec 30 16:51:24 2022 +0530

    first commit
```

```
git reset HEAD~2
```

Undoing Things: go back to a previous commit

```
git status
```

```
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   test.c

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        add_integer.c
```

Soft reset: Staging index and working directory remain as it is.

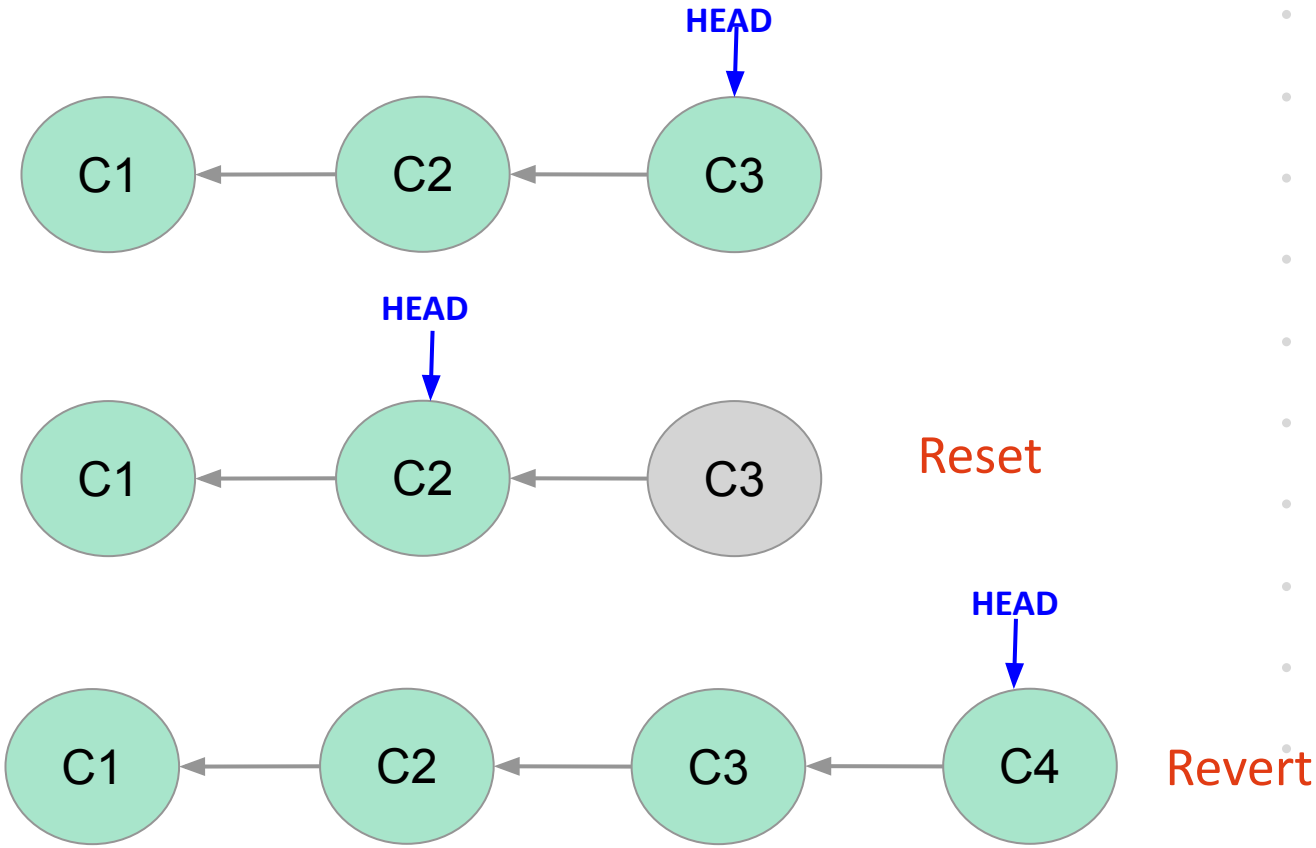
Undoing Things: Hard Reset

```
git reset 44329c79f226226bca9507b90d9ddfc5217f2fe2 --hard
```

```
git status
```

```
On branch master
nothing to commit, working tree clean
```

Undoing Things: Revert



Undoing Things: Revert

```
git revert 413cef719a54ceb372892d3ae13b1aa61c562f91
```

```
commit 75066f20e30a3f5a2f8d0d82f47bbbfcfee38e4a (HEAD -> master)
Author: Joe Cheri Ross <joecheri@gmail.com>
Date:   Mon Jan 9 11:36:07 2023 +0530

    Revert "added add_integer.c to add 2 integers"

    This reverts commit 413cef719a54ceb372892d3ae13b1aa61c562f91.

commit 0eb33bca42d1d82d804e6a42c331540f998ac47a
Author: Joe Cheri Ross <joecheri@gmail.com>
Date:   Wed Jan 4 10:51:17 2023 +0530

    Made a modification to test.c

commit 413cef719a54ceb372892d3ae13b1aa61c562f91
Author: Joe Cheri Ross <joecheri@gmail.com>
Date:   Wed Jan 4 10:50:49 2023 +0530

    added add_integer.c to add 2 integers

commit 44329c79f226226bca9507b90d9ddfc5217f2fe2
Author: Joe Cheri Ross <joecheri@gmail.com>
Date:   Fri Dec 30 16:51:24 2022 +0530

    first commit
```

Removing files

`git rm <path>`

- This removes the file or folder from the tracked file.
- This should be followed by a commit.

```
git rm test.c
```

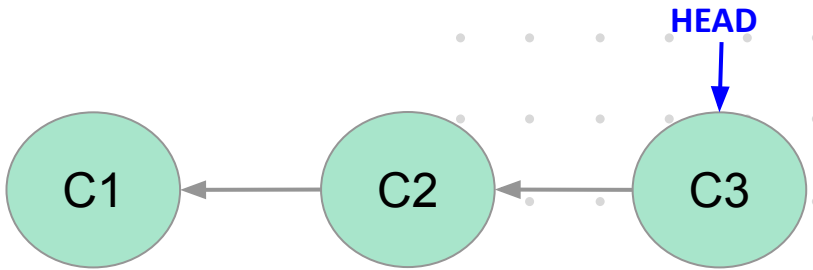
```
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        deleted:    test.c
```

More Insights into Commit

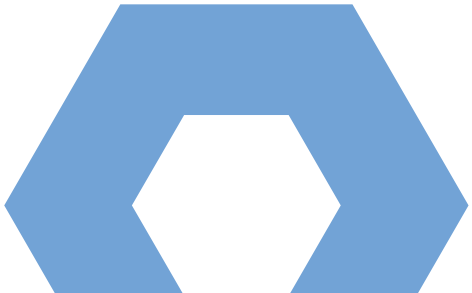
Commit Object

Name
Email
Commit message
Pointer

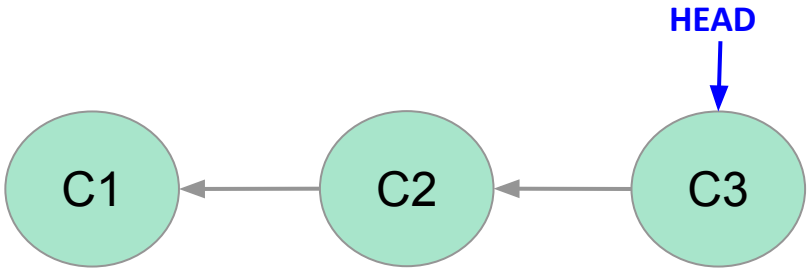
Staging snapshot



Branching



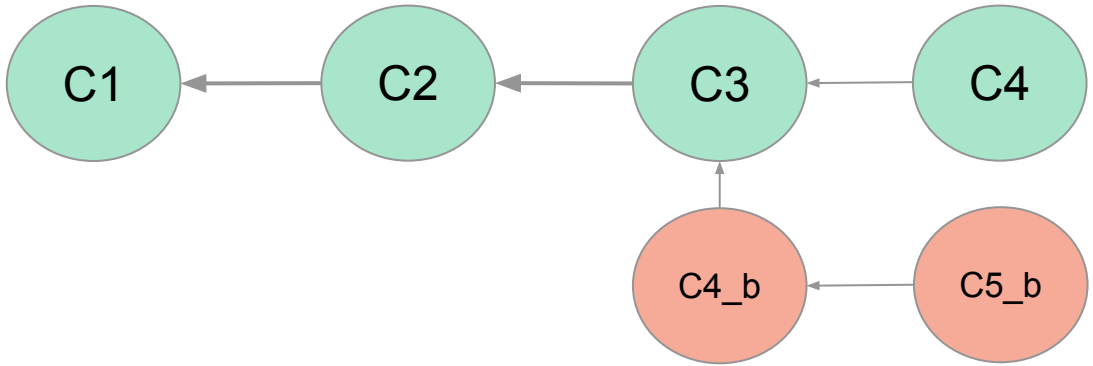
Master Branch




```
On branch master
nothing to commit, working tree clean
```

Why Branching?

Branching helps to diverge from the main line of development without disturbing the main line.



Create a new branch

```
git branch <branch name>
```

```
git branch
```

```
* master
```

```
git branch test_branch
```

```
git branch
```

```
* master
  test_branch
```

Switch to a branch

`git checkout <branch name>`

```
git checkout test_branch
```

```
git branch
```

```
  master
* test_branch
```

Committing in a branch

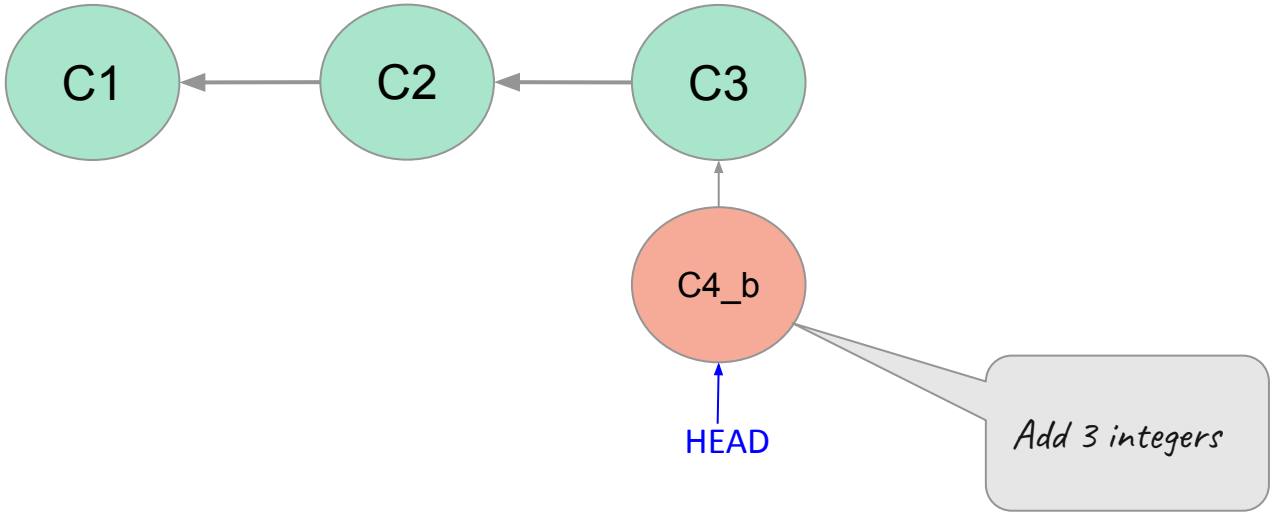
Modified add_integer.c to add 3 integers instead of 2

```
git add add_integer.c
git commit -m "modified add_integer.c to add 3 integers"
```

```
git log
```

```
commit 5ead543a1580d597f4781c3e0200a3869d5332bd (HEAD -> test-branch)
Author: Joe Cheri Ross <joecheri@gmail.com>
Date:   Wed Jan 11 09:33:41 2023 +0530

    modified add_integer.c to add 3 integers

commit ffa0fa22ed1831f0e8cadaaa7278668eb226fe33 (master)
Author: Joe Cheri Ross <joecheri@gmail.com>
Date:   Wed Jan 11 09:28:40 2023 +0530

    modifications to test.c

commit 0bf2e8d800d97824d205eb49635fd6e20c9e96f9
Author: Joe Cheri Ross <joecheri@gmail.com>
Date:   Wed Jan 11 09:27:57 2023 +0530

    added add_integer.c to add 2 integers

commit 614fa030756edebcd781ee9c7140d8feda9cc271
Author: Joe Cheri Ross <joecheri@gmail.com>
Date:   Wed Jan 11 09:26:18 2023 +0530

    first commit
```

Committing to a branch: Visualization



Committing in a branch

```
git checkout master
git log
```

```
commit ffa0fa22ed1831f0e8cadaaa7278668eb226fe33 (HEAD -> master)
Author: Joe Cheri Ross <joecheri@gmail.com>
Date:   Wed Jan 11 09:28:40 2023 +0530

    modifications to test.c

commit 0bf2e8d800d97824d205eb49635fd6e20c9e96f9
Author: Joe Cheri Ross <joecheri@gmail.com>
Date:   Wed Jan 11 09:27:57 2023 +0530

    added add_integer.c to add 2 integers

commit 614fa030756edebcd781ee9c7140d8feda9cc271
Author: Joe Cheri Ross <joecheri@gmail.com>
Date:   Wed Jan 11 09:26:18 2023 +0530

    first commit
```

Committing to a branch

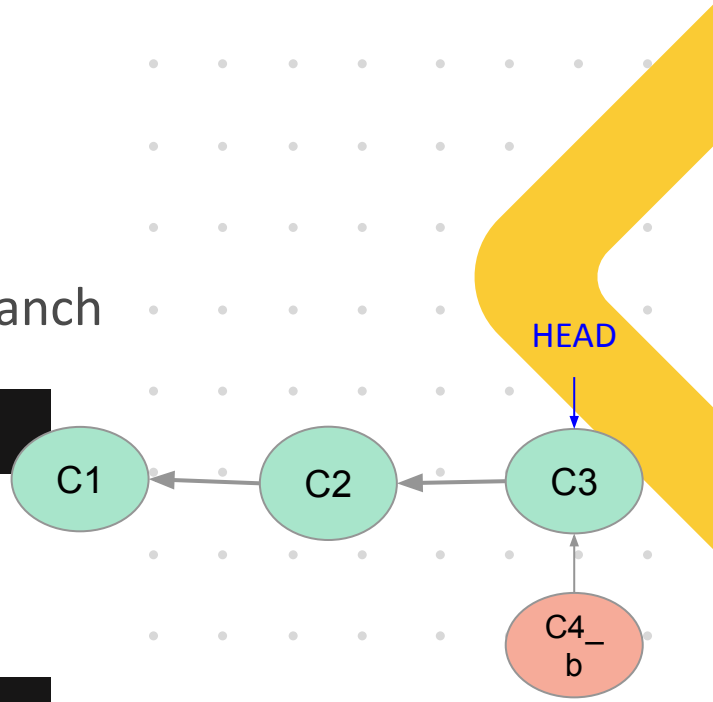
Changes to master branch

Let us make some modifications to the master branch

```
git checkout master
```

Made some changes to test.c

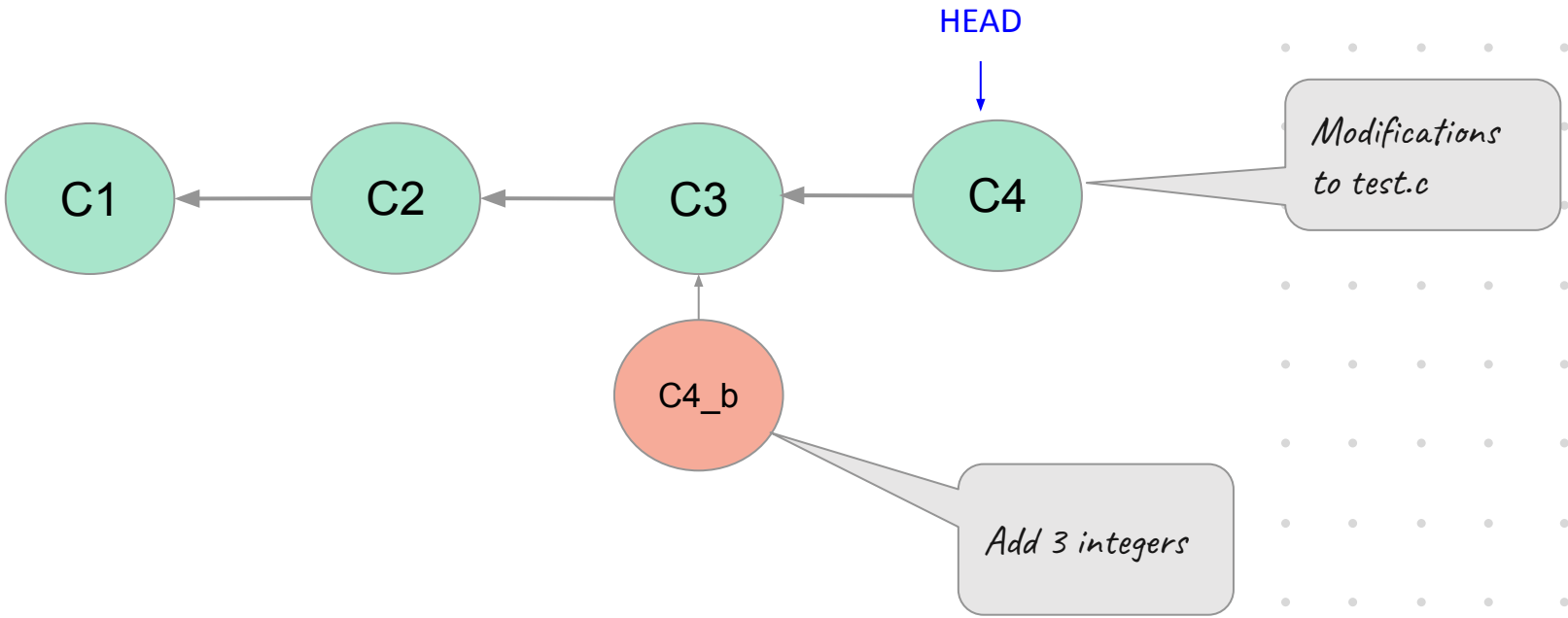
```
git add test.c
git commit -m "some more modifications to test.c"
```



Committing to a branch

Changes to master branch

```
commit e39fe6ad1cc049d52af36b0dc636d1311b60d842 (HEAD -> master)
Author: Joe Cheri Ross <joecheri@gmail.com>
Date:   Wed Jan 11 10:07:31 2023 +0530

    some more modifications to test.c

commit ffa0fa22ed1831f0e8cadaaa7278668eb226fe33
Author: Joe Cheri Ross <joecheri@gmail.com>
Date:   Wed Jan 11 09:28:40 2023 +0530

    modifications to test.c

commit 0bf2e8d800d97824d205eb49635fd6e20c9e96f9
Author: Joe Cheri Ross <joecheri@gmail.com>
Date:   Wed Jan 11 09:27:57 2023 +0530

    added add_integer.c to add 2 integers

commit 614fa030756edebcd781ee9c7140d8feda9cc271
Author: Joe Cheri Ross <joecheri@gmail.com>
Date:   Wed Jan 11 09:26:18 2023 +0530

    first commit
```


HEAD
C1
C2
C3
C4
Modifications to test.c
C4_b
Add 3 integers

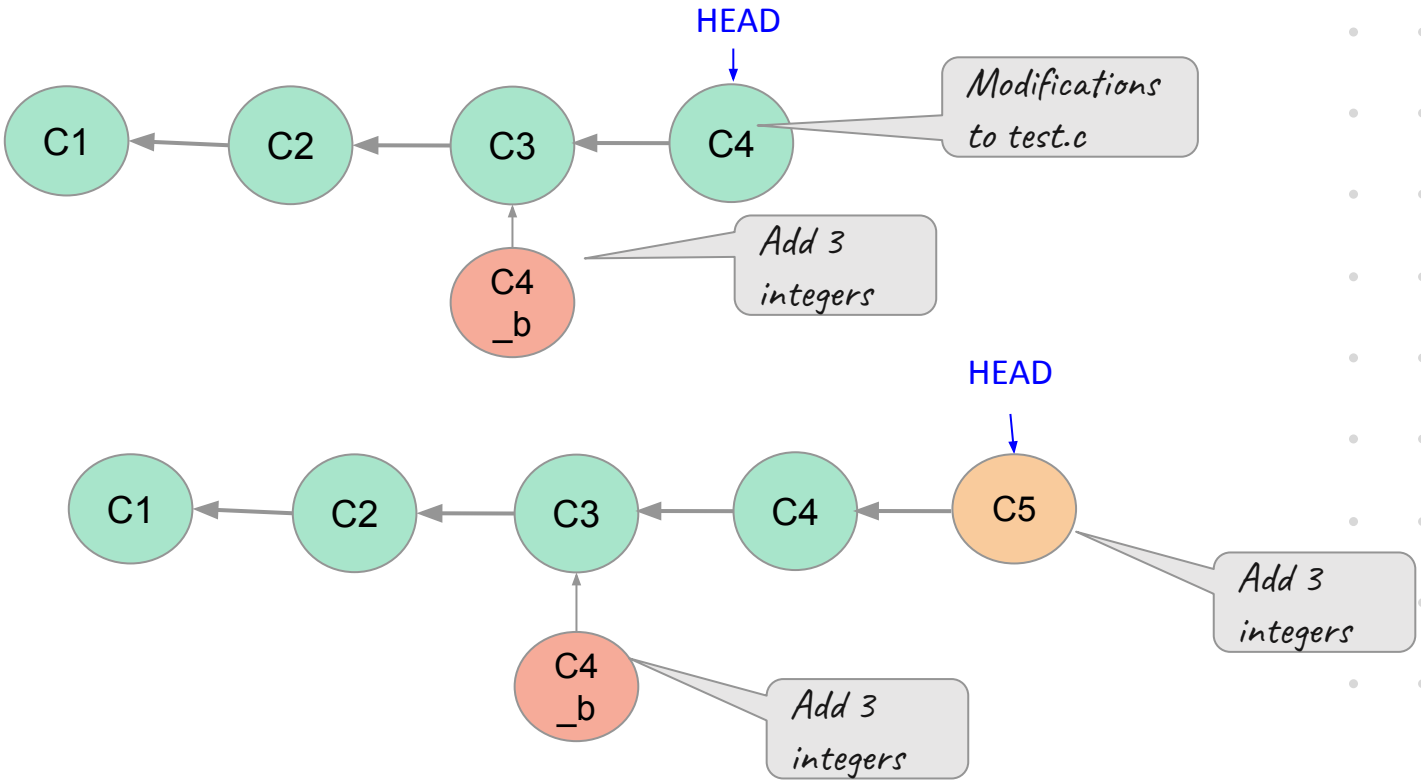
Git log: view all branches

```
git log  --decorate --graph --all --oneline
```

```
* e39fe6a (HEAD -> master) some more modifications to test.c
| * 5ead543 (test-branch) modified add_integer.c to add 3 integers
|/
* ffa0fa2 modifications to test.c
* 0bf2e8d added add_integer.c to add 2 integers
* 614fa03 first commit
```

Merging of branches

`git merge <branch name>`



Merging of branches

```
git checkout master
git merge test-branch
```

```
git log
```

```
*   a282381 (HEAD -> master) Merge branch 'test-branch'
|\
| * 5ead543 (test-branch) modified add_integer.c to add 3 integers
* | e39fe6a some more modifications to test.c
|/
* ffa0fa2 modifications to test.c
* 0bf2e8d added add_integer.c to add 2 integers
* 614fa03 first commit
```