Using git

A workflow suited for 2 (novice) programmers

DISCLAIMER

This is by no means a complete guide - far from it actually.

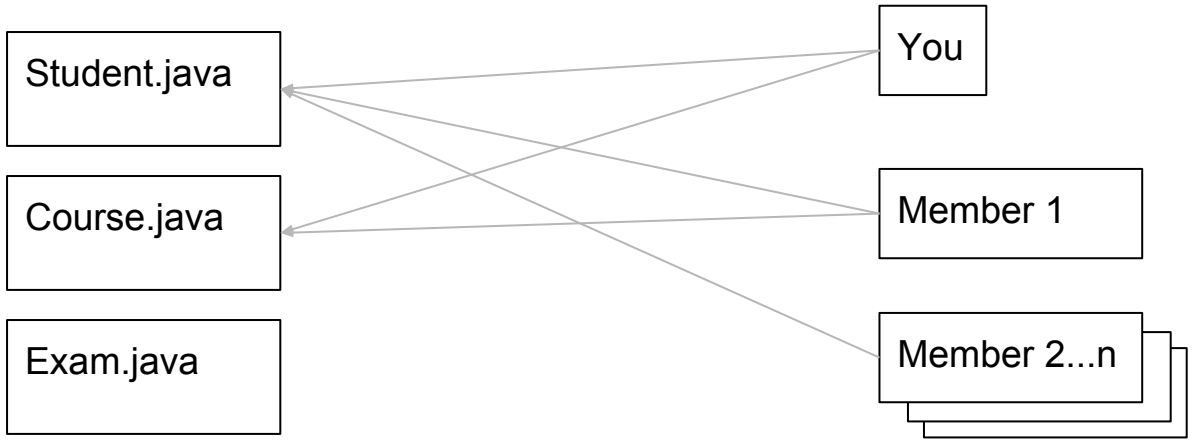
It is quick tour through the basics of git and provides a workflow suited for students not used to working with VCS.

We are no git experts!

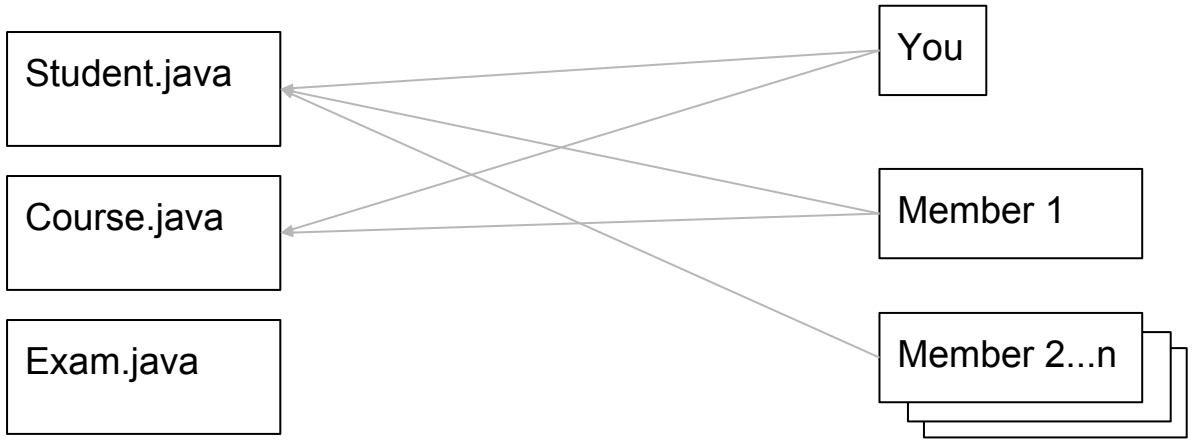
If we all pull together as a team

Imagine this

Imagine this



Merging the changes - who?



Merging the changes - who and how?

Dropbox	no
Zip files	no
Email files	no

Use only one computer	works, nice if two but does not scale

VCS - to the rescue .... And a bit of history

VCS - Versioning Control System

- A program to manage files in your project.
- Changes are stored as different versions of your files.

RCS	1982
CVS	1990
SVN	2000
Darcs	2003
Mercurial	2005
Bazaar	2005
Git	2005

Install git

Install using

- MacOS: Homebrew or MacPorts
- Windows: cygwin
- Ubuntu | Fedora: apt-get | dnf

.. and then

```
$ git config --global user.name "John Doe"

$ git config --global user.email "john@example.com"
```

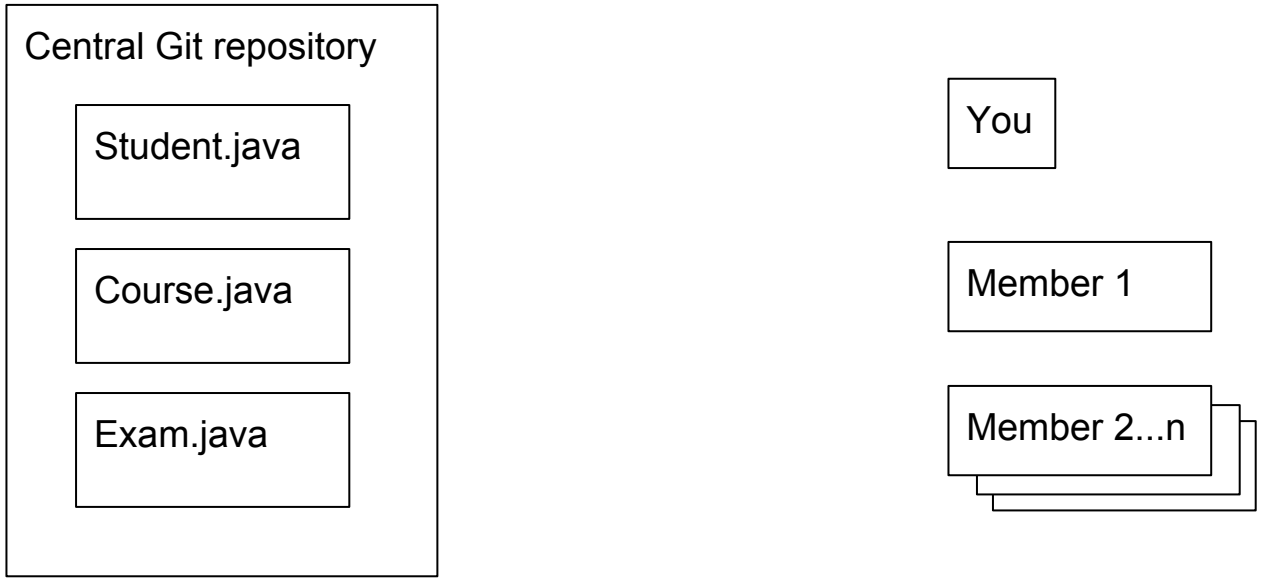
VCS - git

Free git services

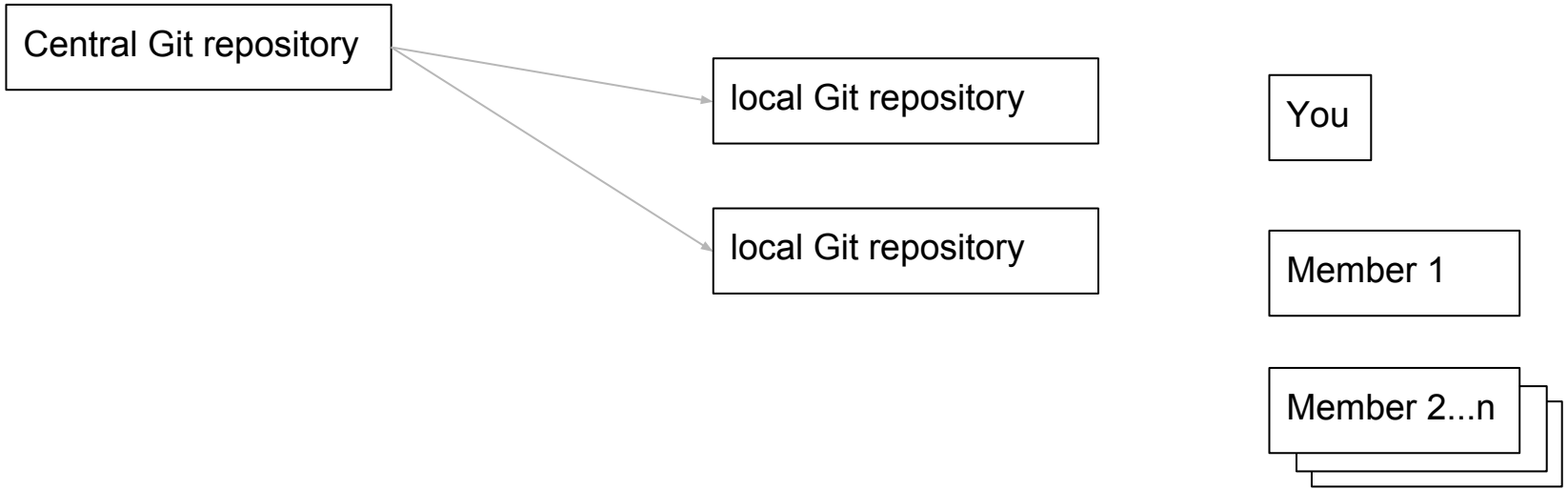
- gitlab.com
- github.com
- bitbucket.com
- savannah.nongnu.org
- sourceforge.com
- ……

Create an account, add ssh keys, create two repositories (client and server)

Git


Central Git repository
Student.java
Course.java
Exam.java
You
Member 1
Member 2...n

Git - clone repository



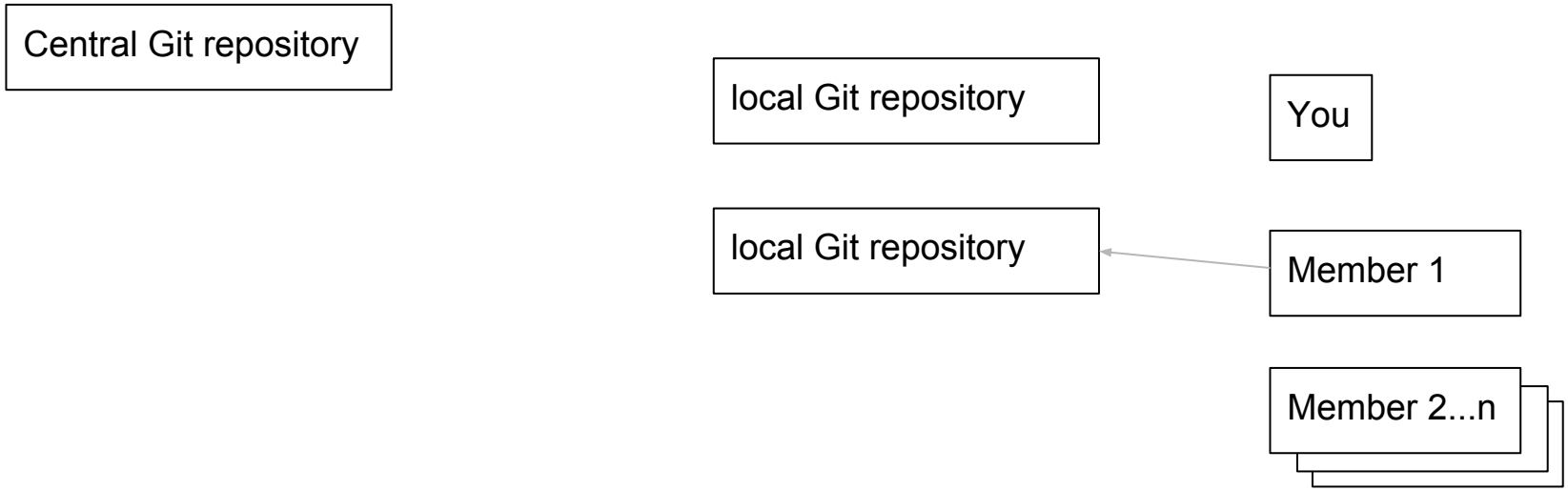

```
$ git clone git@github.com:ozzy/lemmy.git
```

Git - edit them files

Git - commit your edits (local repository)

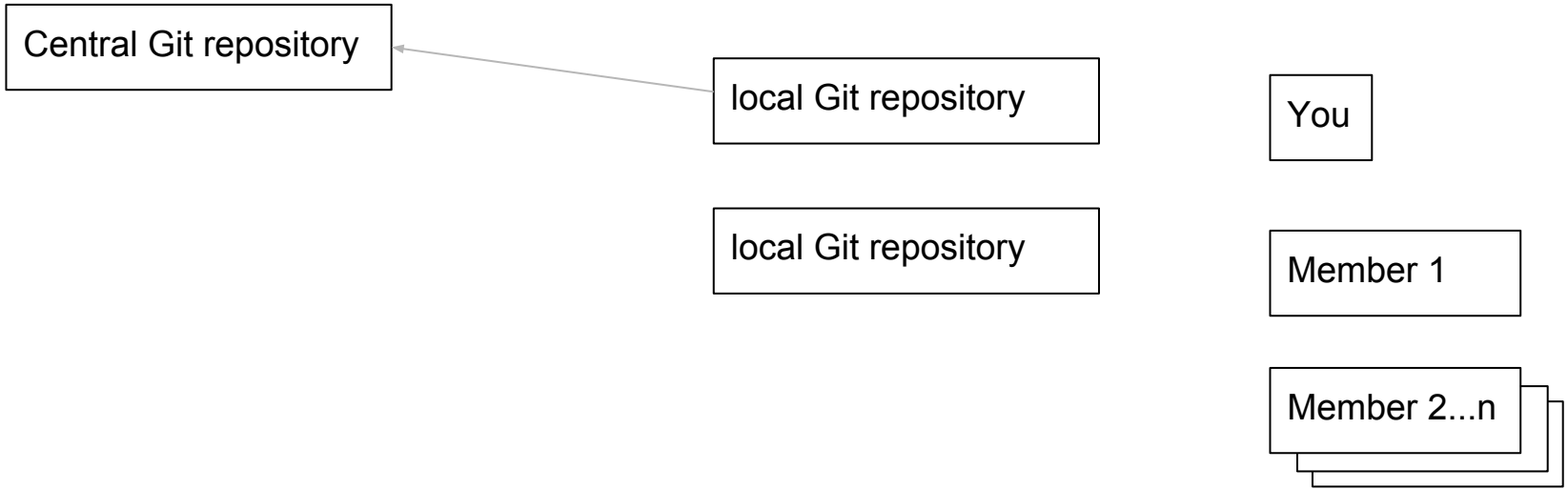

```
$ git add Member.java
$ git commit -m "fixed typo" Member.java


$ git commit -m "Email addresses are checked"
```

Git - commit your edits (local repository)



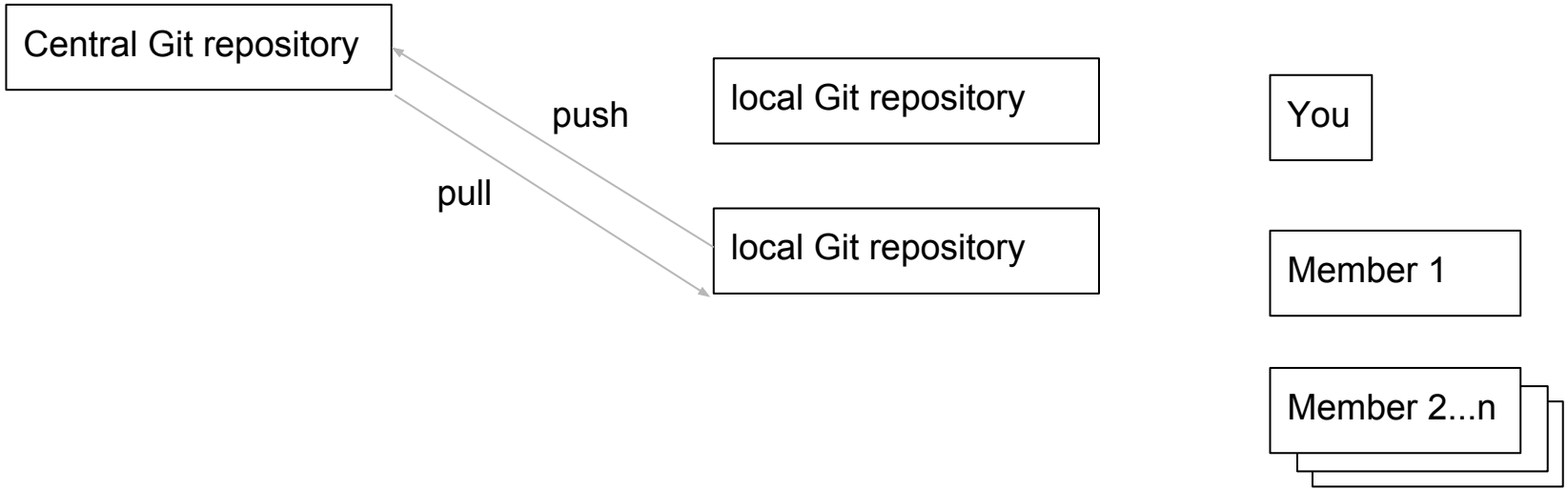

```
$ git add Member.java
$ git commit -m "added email support" Member.java
```

Git - push your edits




```
$ git push
```

Git - push your edits - uh oh, out of sync




```
$ git pull
# possibly merge files
$ git push
```

Git useful commands

`clone`	- clone a repository
`add`	- prepare content to commit
`commit`	- write changes to your local repo
`push`	- upload your changes to the central repo
`pull`	- download the latest version (and merge)
`status`	- status of the working tree
`log`	- show commit logs

Some more commands

`init`	- create an empty repository
`diff`	- compare two files (between commits)
`status`	- show the status of the “working tree”
`checkout`	- used to restore file

Live video

Create repository

Two users (in two terminal):

- `clone`
- Edit files
- `add` and `commit`
- `push`
- `pull` and `merge`
- `Diff`
- `log`
- `status`
- `checkout`