

Using Git

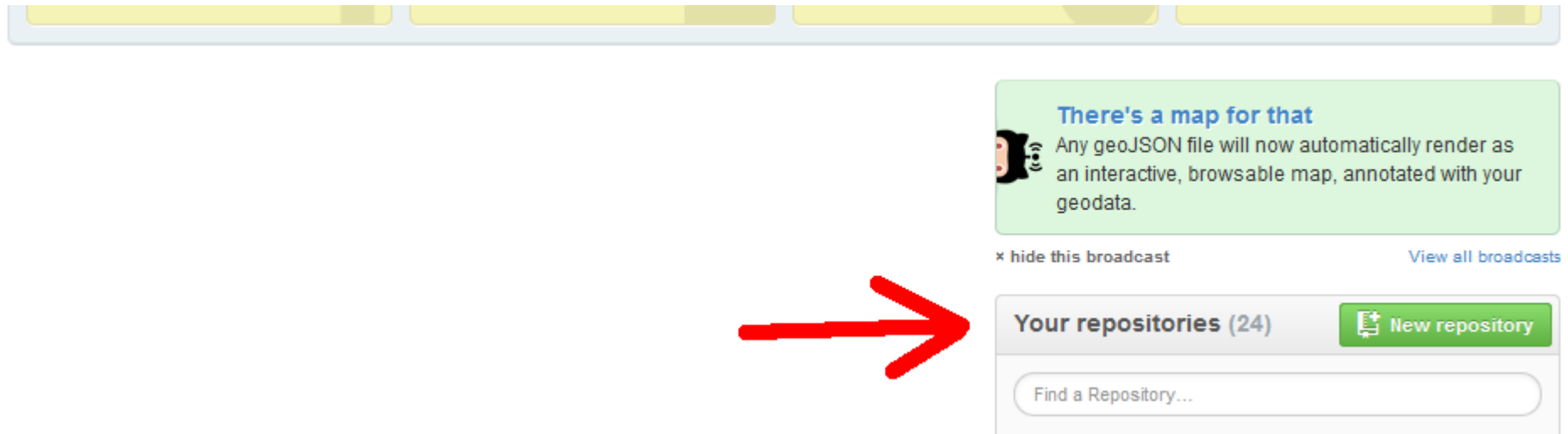
With GitBash / CLI (Command Line Interface)

Available anywhere Git is installed!

- Creating a new Project Repository
- Sharing the Repository with the instructor(s)
- Downloading Repository from server to harddrive
- Adding new files
- Creating a commit
- Pushing to/Syncing with the server
- Pulling new changes

1. Creating a new Project Repository




- From <https://github.com/>, scroll down until you see the green New repository button under the Your repositories section. Click the button.



1. Creating a new Project Repository

- On the Create a New Repository page, fill out the following:
 - Repository Name: CS201
 - Description: Optional
 - **Select Private** radio button
 - If it asks for a credit card, you haven't been activated as a student account yet. See week 1 slides
 - **Check** Initialize this repository with a README
 - From the dropdown box, select C++ as the language


1. Creating a new Project Repository

 Owner:  rjmfff / Repository name: CS201L 


Great repository names are short and memorable. Need inspiration? How about **hairy-octo-tribble**.

Description (optional)

☐  **Public**
Anyone can see this repository. You choose who can commit.

☒  **Private**
You choose who can see and commit to this repository.

☒ **Initialize this repository with a README**
This will allow you to `git clone` the repository immediately.

Add .gitignore: C++ 

Create repository

1. Creating a new Project Repository

- You will need to create an additional repository if you're in the CS 201 Lab.
 - CS 201
 - CS 201 L

2. Sharing your Repositories

- For CS 201, you will need to share your repository with Rachel.
- For CS 201 L, you will need to share your repository with Imrul and Rachel.

2. Sharing your Repositories

- From your Repository website,
https://github.com/YOUR_USERNAME/CS201
Click on the Settings button.

2. Sharing your Repositories

The screenshot shows a GitHub repository page for 'rjmfff / CS201'. The repository is marked as 'PRIVATE' with a lock icon. At the top, there are buttons for 'Pull Request', 'Unwatch', 'Star' (0), and 'Fork' (0). Below these are tabs for 'Code', 'Network', 'Pull Requests' (0), 'Issues' (0), 'Wiki', 'Graphs', and 'Settings'. A red arrow points to the 'Settings' tab. The 'Code' tab is active, showing a 'No description or homepage.' message. Below this are buttons for 'Clone in Windows', 'ZIP', and 'HTTP/SSH' links. The SSH link is 'https://github.com/rjmfff/CS201.git'. A 'Read+Write access' button is also visible. Below the clone buttons, there is a 'branch: master' dropdown and tabs for 'Files', 'Commits', 'Branches' (1), and 'Tags'. The 'Files' tab is active, showing a list of files: '.gitignore' and 'README.md', both committed 'a minute ago' by 'rjmfff'. The commit hash 'ed48ed41ae' is shown. Below the file list, there is a 'README.md' section.

PRIVATE rjmfff / CS201

Pull Request Unwatch Star 0 Fork 0

Code Network Pull Requests 0 Issues 0 Wiki Graphs Settings

No description or homepage.

Clone in Windows ZIP HTTP SSH https://github.com/rjmfff/CS201.git Read+Write access

branch: master Files Commits Branches 1 Tags

CS201 / 1 commit

Initial commit

rjmfff authored a minute ago latest commit ed48ed41ae

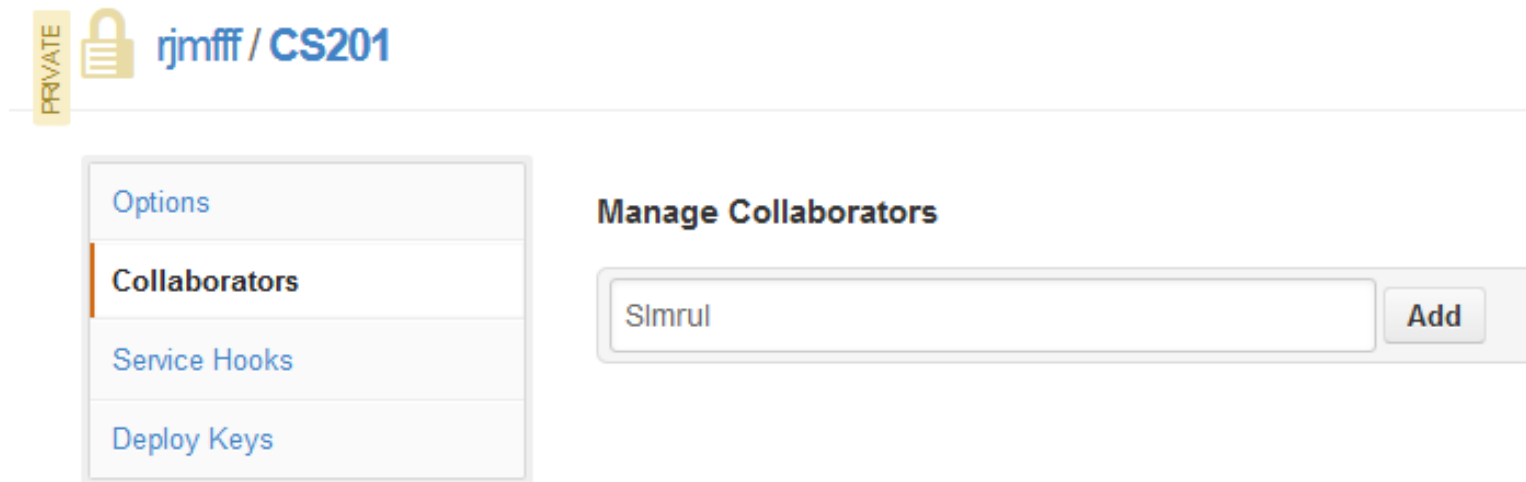
.gitignore	a minute ago	Initial commit [rjmfff]
README.md	a minute ago	Initial commit [rjmfff]

README.md

2. Sharing your Repositories

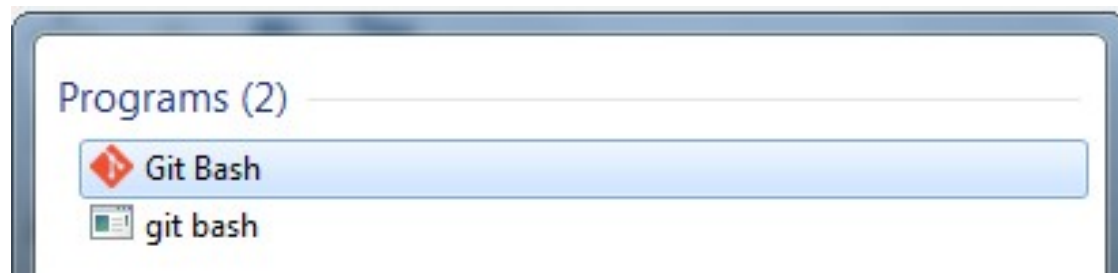
- From the Settings page, click on the Collaborators tab.
- In the Collaborators tab, enter our IDs:
 - RJMFFF for Rachel (CS 201 and CS 201 L)
 - Simrul for Imrul
 - Usernames are not case-sensitive
- Click on the Add button.

2. Sharing your Repositories



3. Setting up Git

- If you have GitHub for Windows/Mac installed or GitGUI installed, then you are able to use Git from the Command Line as well!
- CLI stands for “Command Line Interface”.
- Open:
 - Git Bash from the Start Menu (Windows)
 - Open a terminal (*NIX)

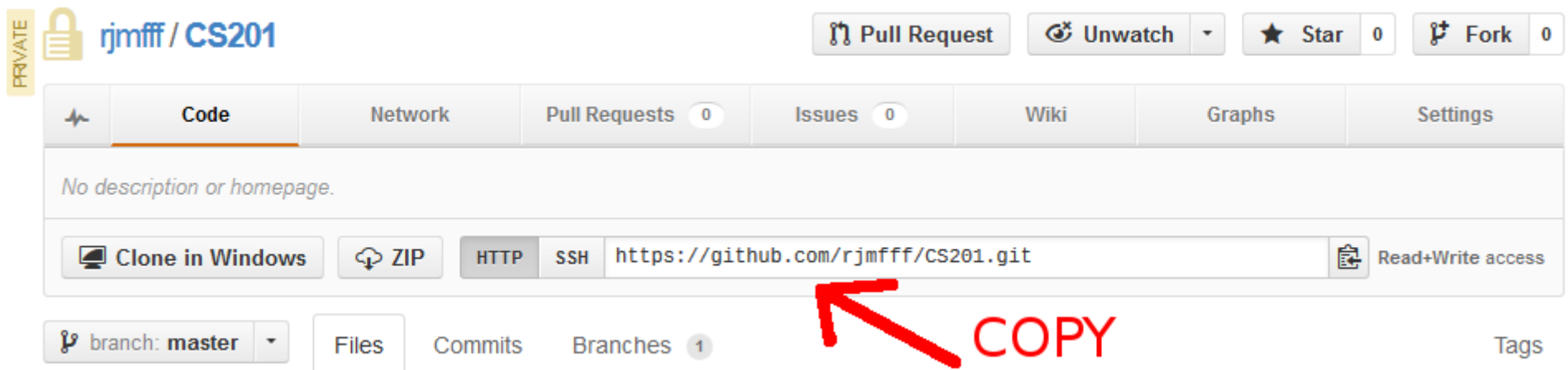


4. Downloading your Repository

- The repository is currently only on the web server. To add and change files, you need to pull it down to your local machine.
- You will modify files on your harddrive, and when you're done with a change, push them back to the server.

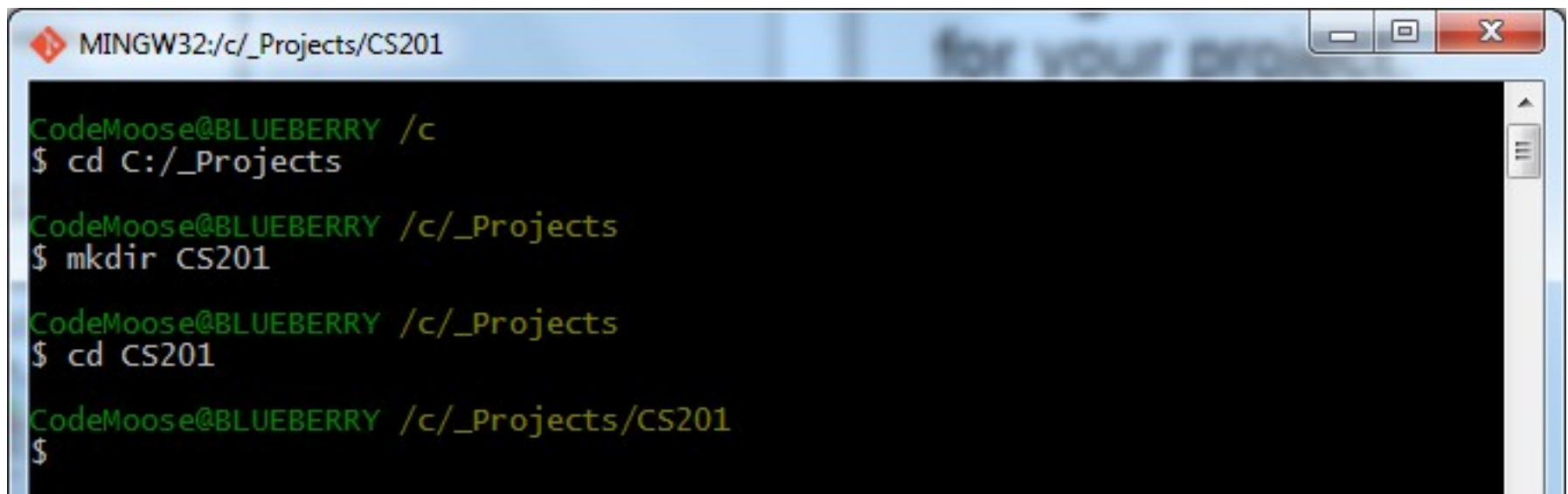
4. Downloading your Repository

- From your Repository website, https://github.com/YOUR_USERNAME/CS201 highlight the .git URL and copy it.



4. Downloading your Repository

- Navigate Git Bash or your terminal to a folder for your project.

A screenshot of a Git Bash terminal window. The title bar at the top reads 'MINGW32:/c/_Projects/CS201'. The terminal has a black background with green text. It shows a series of commands and their outputs: the prompt 'CodeMoose@BLUEBERRY /c' followed by '\$ cd C:/_Projects', then 'CodeMoose@BLUEBERRY /c/_Projects' followed by '\$ mkdir CS201', then 'CodeMoose@BLUEBERRY /c/_Projects' followed by '\$ cd CS201', and finally 'CodeMoose@BLUEBERRY /c/_Projects/CS201' followed by '\$'.

```
MINGW32:/c/_Projects/CS201

CodeMoose@BLUEBERRY /c
$ cd C:/_Projects

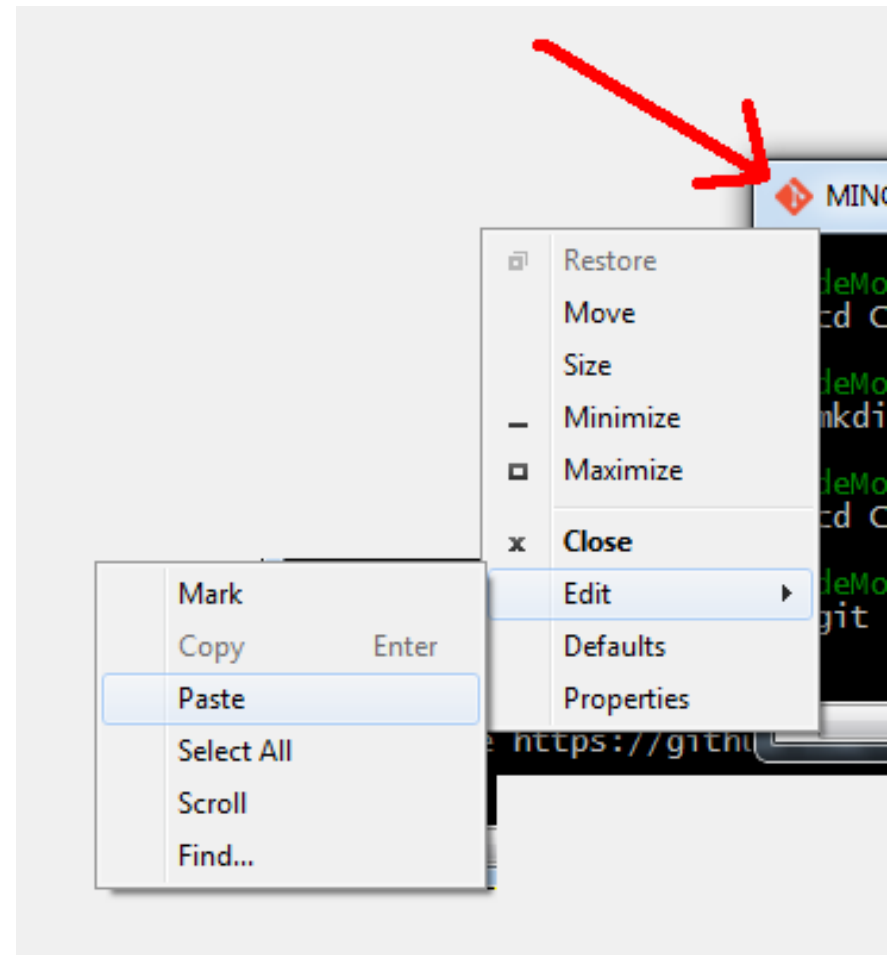
CodeMoose@BLUEBERRY /c/_Projects
$ mkdir CS201

CodeMoose@BLUEBERRY /c/_Projects
$ cd CS201

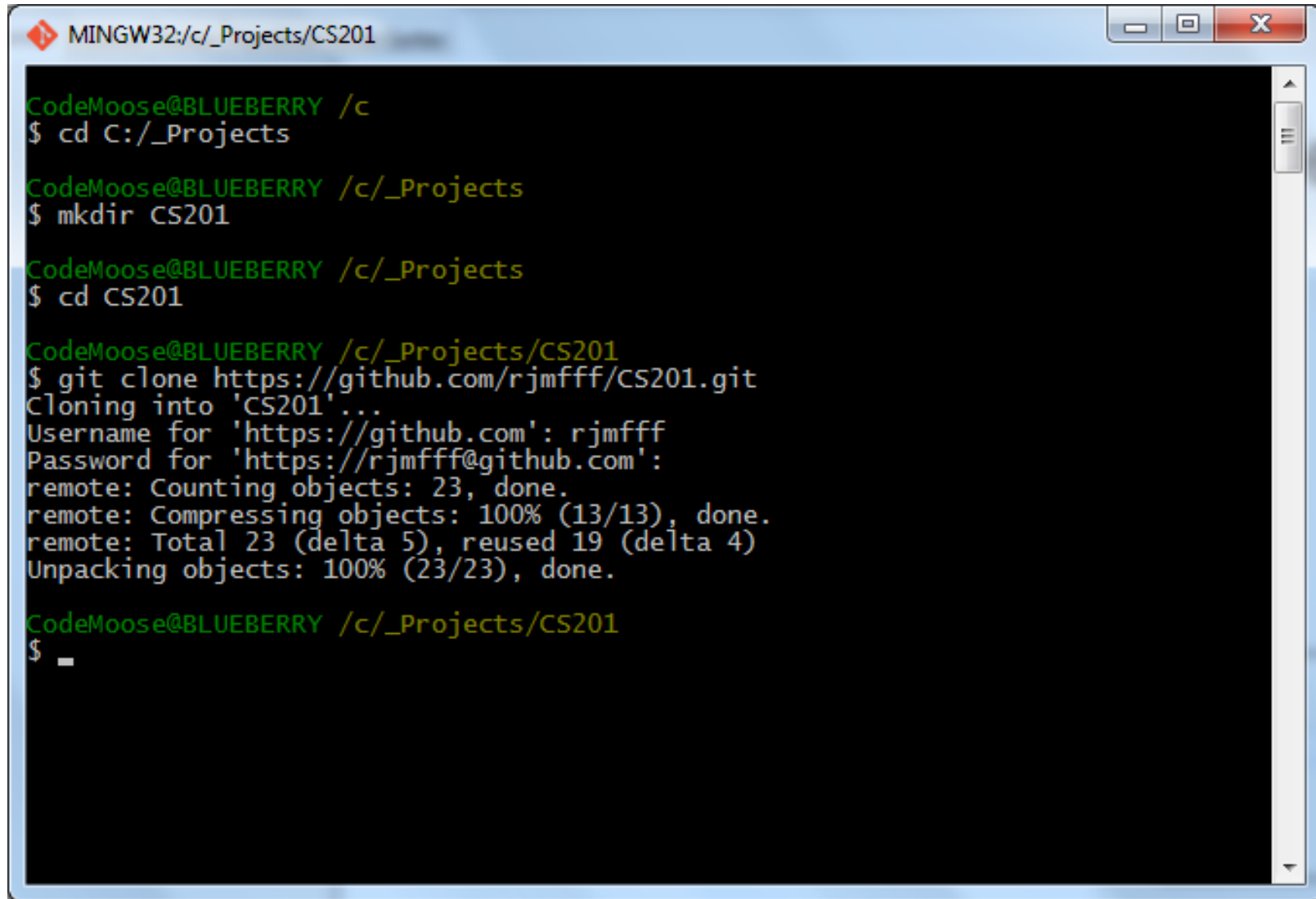
CodeMoose@BLUEBERRY /c/_Projects/CS201
$
```

4. Downloading your Repository

- Type “git clone”, then paste in the .git URL.
- In GitBash, you can paste by clicking on the program icon, clicking Edit, then Paste.
- Hit enter.



4. Downloading your Repository



```
MINGW32:/c/_Projects/CS201

CodeMoose@BLUEBERRY /c
$ cd C:/_Projects

CodeMoose@BLUEBERRY /c/_Projects
$ mkdir CS201

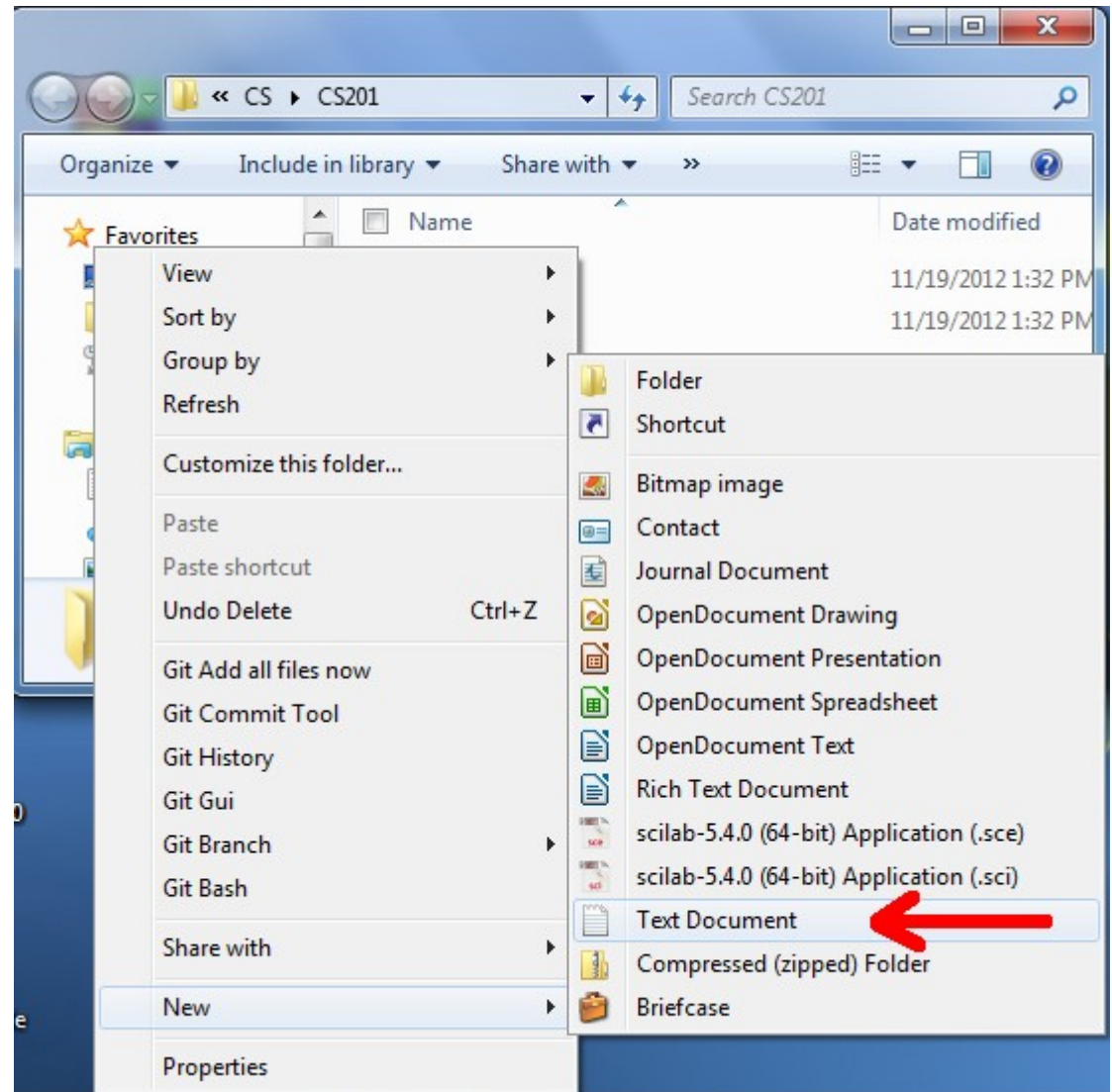
CodeMoose@BLUEBERRY /c/_Projects
$ cd CS201

CodeMoose@BLUEBERRY /c/_Projects/CS201
$ git clone https://github.com/rjmfff/CS201.git
Cloning into 'CS201'...
Username for 'https://github.com': rjmfff
Password for 'https://rjmfff@github.com':
remote: Counting objects: 23, done.
remote: Compressing objects: 100% (13/13), done.
remote: Total 23 (delta 5), reused 19 (delta 4)
Unpacking objects: 100% (23/23), done.

CodeMoose@BLUEBERRY /c/_Projects/CS201
$ _
```

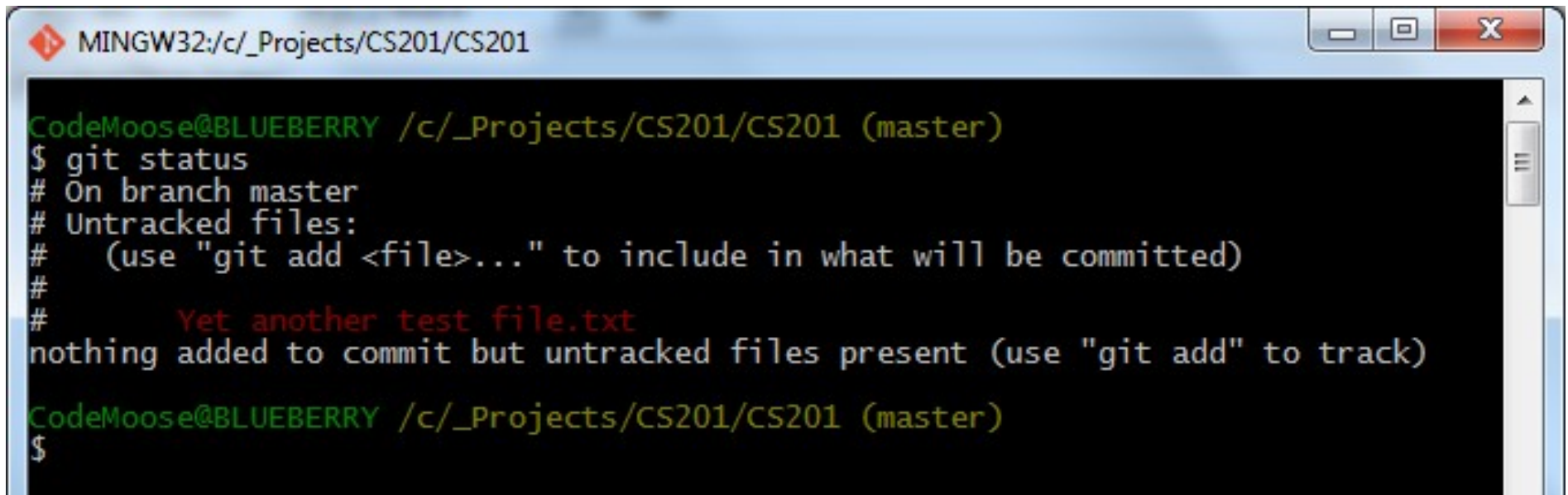

5. Adding a test file

- Navigate to your project folder in Windows Explorer (or whatever).
- Create a new text document in the repository folder.



5. Adding a test file

- From Git Bash, type:
git status
- This tells you what files have been changed.



```
MINGW32:/c/_Projects/CS201/CS201

CodeMoose@BLUEBERRY /c/_Projects/CS201/CS201 (master)
$ git status
# On branch master
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#       Yet another test file.txt
nothing added to commit but untracked files present (use "git add" to track)
CodeMoose@BLUEBERRY /c/_Projects/CS201/CS201 (master)
$
```

5. Adding a test file

- You can add a specific file by typing:
`git add [filename]`
- Or add all changes by typing:
`git add *`
- If you type the first few characters of a filename and hit TAB, it will try to autocomplete for you.
- If there are spaces in the filename, you need to prefix the spaces with `\` or include the filename within double-quotes.

5. Adding a test file

- Use git add to add the new file.

```
CodeMoose@BLUEBERRY /c/_Projects/CS201/CS201 (master)
$ git add Yet\ another\ test\ file.txt

CodeMoose@BLUEBERRY /c/_Projects/CS201/CS201 (master)
$ git add "Yet another test file.txt"

CodeMoose@BLUEBERRY /c/_Projects/CS201/CS201 (master)
$ git add *
```

5. Adding a test file

git status before add:

```
CodeMoose@BLUEBERRY /c/_Projects/CS201/CS201 (master)
$ git status
# On branch master
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#       Yet another test file.txt
nothing added to commit but untracked files present (use "git add" to track)
```

git status after add:

```
CodeMoose@BLUEBERRY /c/_Projects/CS201/CS201 (master)
$ git add *

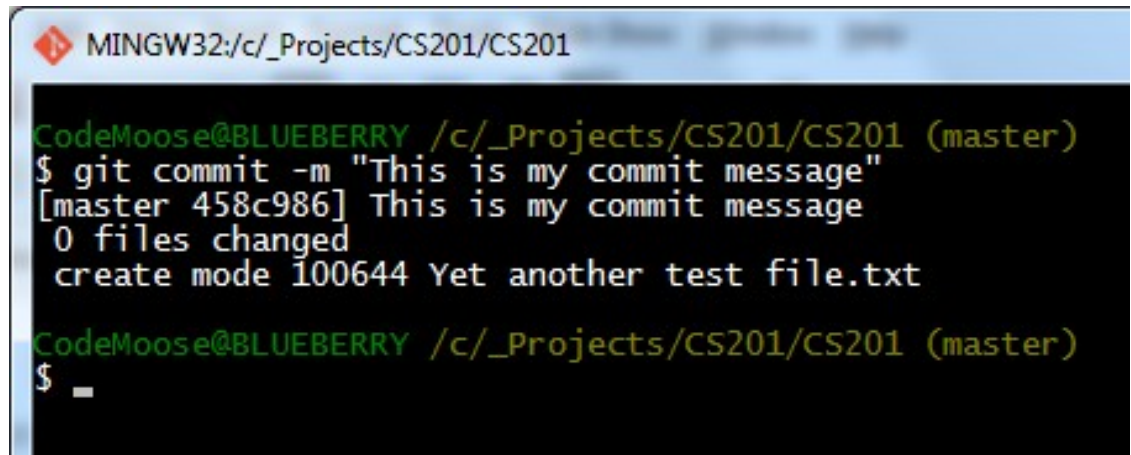
CodeMoose@BLUEBERRY /c/_Projects/CS201/CS201 (master)
$ git status
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#       new file:   Yet another test file.txt
#
```

5. Adding a test file

- Now we need to commit the file we added. A commit can contain multiple files and has a label message so you can keep track of your changes over time.
- You can commit with either:
git commit
or
git commit -m "This is my commit message"

5. Adding a test file

- If you use `git commit` (without the `-m` flag), your OS will open a text editor for you to add the commit message.
- If it opens Vim... well, I don't know how to use Vim. Close the bash and commit with the `-m` flag. :P

A screenshot of a Windows command prompt window titled 'MINGW32:/c/_Projects/CS201/CS201'. The prompt shows a user named 'CodeMoose@BLUEBERRY' in the directory '/c/_Projects/CS201/CS201' on the 'master' branch. The user enters the command '\$ git commit -m "This is my commit message"'. The output shows '[master 458c986] This is my commit message', '0 files changed', and 'create mode 100644 Yet another test file.txt'. The prompt then shows the user entering '\$' and a cursor, indicating the command has finished and the prompt is ready for the next input.

```
MINGW32:/c/_Projects/CS201/CS201

CodeMoose@BLUEBERRY /c/_Projects/CS201/CS201 (master)
$ git commit -m "This is my commit message"
[master 458c986] This is my commit message
0 files changed
create mode 100644 Yet another test file.txt

CodeMoose@BLUEBERRY /c/_Projects/CS201/CS201 (master)
$
```

6. Pushing your changes to the server

- Now we've committed our changes, but it isn't backed up to the server until we actually push the changes.
- To push our list of commits to the server, use `git push -u origin master`
- Don't worry about what “-u origin master” means for this class, just memorize it. (It has to do with branching).

6. Pushing your changes to the server

```
CodeMoose@BLUEBERRY /c/_Projects/CS201/CS201 (master)
$ git push -u origin master
Username for 'https://github.com': rjmfff
Password for 'https://rjmfff@github.com':
Counting objects: 4, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 303 bytes, done.
Total 3 (delta 1), reused 0 (delta 0)
To https://github.com/rjmfff/CS201.git
    160186d..458c986  master -> master
Branch master set up to track remote branch master from origin.

CodeMoose@BLUEBERRY /c/_Projects/CS201/CS201 (master)
$
```

6. Pushing your changes to the server

- Your changes will now be visible on the webpage.

The screenshot shows the GitHub interface for the repository 'rjmfff / CS201'. At the top, there are buttons for 'Pull Request', 'Unwatch', 'Star' (0), and 'Fork' (0). Below these are tabs for 'Code', 'Network', 'Pull Requests' (0), 'Issues' (0), 'Wiki', 'Graphs', and 'Settings'. The 'Code' tab is selected, showing a 'No description or homepage.' message. Below this are buttons for 'Clone in Windows', 'ZIP', and 'HTTP/SSH' links. The SSH link is 'https://github.com/rjmfff/CS201.git'. To the right is a 'Read+Write access' button. Below the clone buttons is a 'branch: master' dropdown and tabs for 'Files', 'Commits', 'Branches' (1), and 'Tags'. The 'Commits' tab is selected, showing a list of commits. The first commit is by 'Moosader' 2 minutes ago, with the message 'This is my commit message'. The commit hash is '458c986407'. Below the commit list are several files: '.gitattributes' (4 days ago), '.gitignore' (42 minutes ago), 'New Text File.txt' (an hour ago), 'README.md' (an hour ago), 'Testing File.txt' (25 minutes ago), and 'Yet another test file.txt' (2 minutes ago). Each file has a corresponding commit message link.

CS201 / [+](#) 8 commits

This is my commit message

Moosader authored 2 minutes ago latest commit 458c986407

File	Time	Commit Message
.gitattributes	4 days ago	Test Commit [Moosader]
.gitignore	42 minutes ago	Merge? [Moosader]
New Text File.txt	an hour ago	Created a new test file [Moosader]
README.md	an hour ago	Initial commit [rjmfff]
Testing File.txt	25 minutes ago	This is my test file! [Moosader]
Yet another test file.txt	2 minutes ago	This is my commit message [Moosader]