Git and Source Control

Written by Rachel J. Morris, last updated Jan 10, 2018

About

Source Control is an important tool for any software developer and it is important to know how to use it.

Topics

What is source control?

1. What is source control?

Source Control is also sometimes known as Version Control or Revision Control.

It is a tool that is (or should be) used by all developers, and by all companies that write code.

Notes

1. What is source control?

The general idea is:

- It keeps track of your changes over time.
- It makes it easier to merge code together (whether between multiple people, or if you’re working on different machines)
- It makes code easier to share
- Code is saved on a server, so your work shouldn’t get lost.

Notes

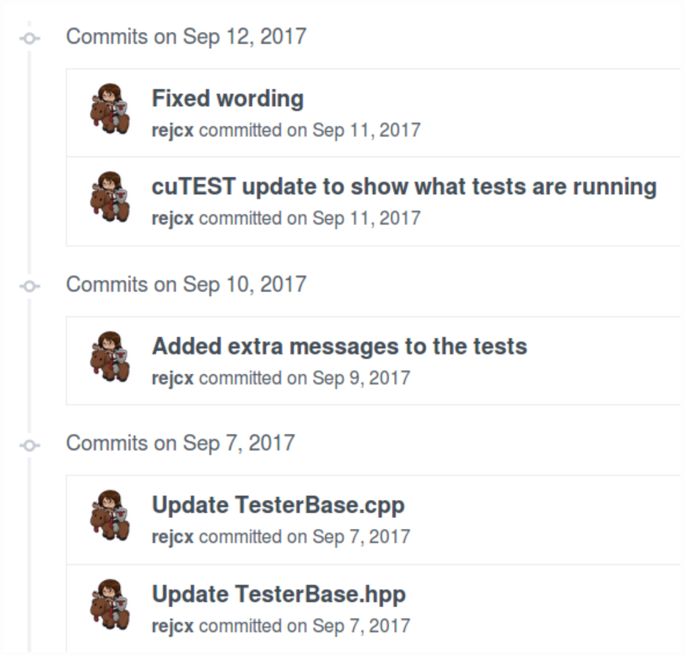
Source Control:

- Save changes to server
- Keep track of changes over time
- Merge code together
- Share code

Example of features

2. Features: Keeping track of changes

When you have a smart system keeping track of changes over time, you can view all the changes throughout history…


Commits on Sep 12, 2017

Fixed wording
rejcx committed on Sep 11, 2017

cuTEST update to show what tests are running
rejcx committed on Sep 11, 2017

Commits on Sep 10, 2017

Added extra messages to the tests
rejcx committed on Sep 9, 2017

Commits on Sep 7, 2017

Update TesterBase.cpp
rejcx committed on Sep 7, 2017

Update TesterBase.hpp
rejcx committed on Sep 7, 2017

List of changes made

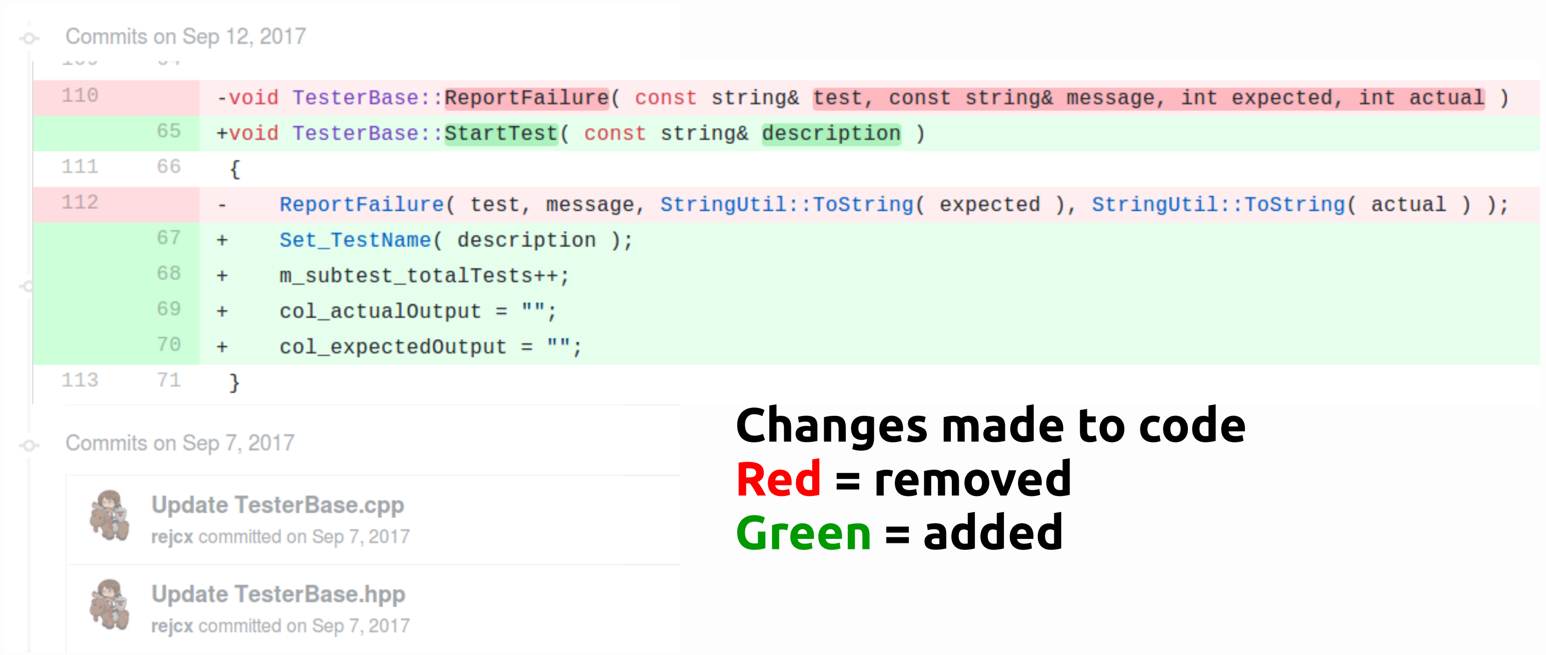
Notes

Source Control:

- Save changes to server
- Keep track of changes over time
- Merge code together
- Share code

2. Features: Keeping track of changes

When you have a smart system keeping track of changes over time, you can view all the changes throughout history…



Commits on Sep 12, 2017

```
-void TesterBase::ReportFailure( const string& test, const string& message, int expected, int actual )
+void TesterBase::StartTest( const string& description )
 {
-    ReportFailure( test, message, StringUtil::ToString( expected ), StringUtil::ToString( actual ) );
+    Set_TestName( description );
+    m_subtest_totalTests++;
+    col_actualOutput = "";
+    col_expectedOutput = "";
 }
```

Commits on Sep 7, 2017

Update TesterBase.cpp
rejcx committed on Sep 7, 2017

Update TesterBase.hpp
rejcx committed on Sep 7, 2017

Changes made to code
Red = removed
Green = added

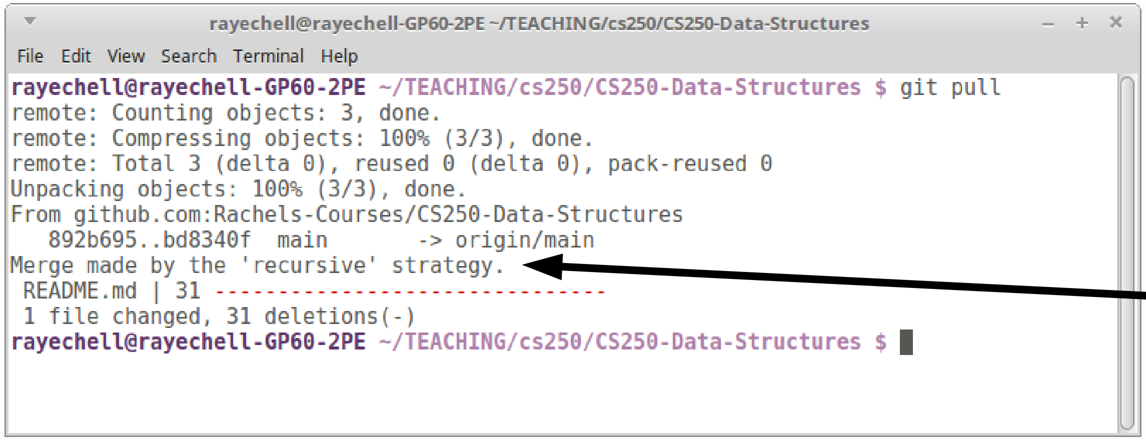
Notes

Source Control:

- Save changes to server
- Keep track of changes over time
- Merge code together
- Share code

2. Features: Merging

If you're moving between computers, or working on the same code file at the same time as someone else, a Source Control system will merge the files as best as it can…


rayechell@rayechell-GP60-2PE ~/TEACHING/cs250/CS250-Data-Structures
File Edit View Search Terminal Help

```
rayechell@rayechell-GP60-2PE ~/TEACHING/cs250/CS250-Data-Structures $ git pull
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From github.com:Rachels-Courses/CS250-Data-Structures
   892b695..bd8340f  main       -> origin/main
Merge made by the 'recursive' strategy.
 README.md | 31 -------------------------------
 1 file changed, 31 deletions(-)
rayechell@rayechell-GP60-2PE ~/TEACHING/cs250/CS250-Data-Structures $ 
```

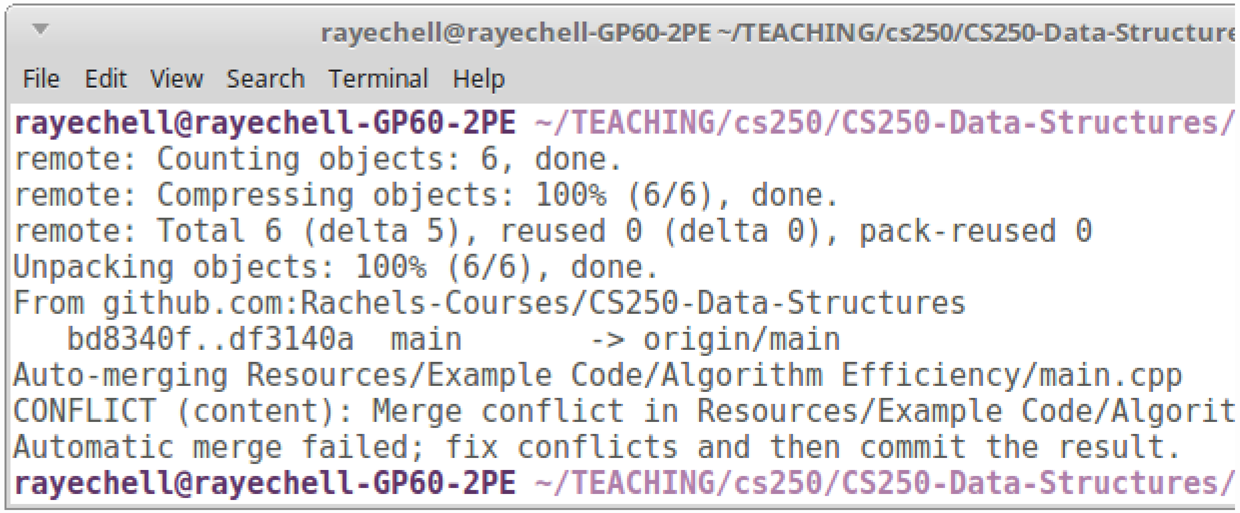
It automatically merged changes between the local machine and the changes on the server for "README.md".

Notes

Source Control:
- Save changes to server
- Keep track of changes over time
- Merge code together
- Share code

2. Features: Merging

And if it can't figure out how to merge (usually if changes are made to the same region of the same file), then it will add markers to the file so you can manually merge as you see fit. This is easier than trying to read two files and figure out where the changes are yourself.


rayechell@rayechell-GP60-2PE ~/TEACHING/cs250/CS250-Data-Structure
File Edit View Search Terminal Help

```
rayechell@rayechell-GP60-2PE ~/TEACHING/cs250/CS250-Data-Structures/
remote: Counting objects: 6, done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 6 (delta 5), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (6/6), done.
From github.com:Rachels-Courses/CS250-Data-Structures
   bd8340f..df3140a  main       -> origin/main
Auto-merging Resources/Example Code/Algorithm Efficiency/main.cpp
CONFLICT (content): Merge conflict in Resources/Example Code/Algorit
Automatic merge failed; fix conflicts and then commit the result.
rayechell@rayechell-GP60-2PE ~/TEACHING/cs250/CS250-Data-Structures/
```

It tells you there's a merge conflict first

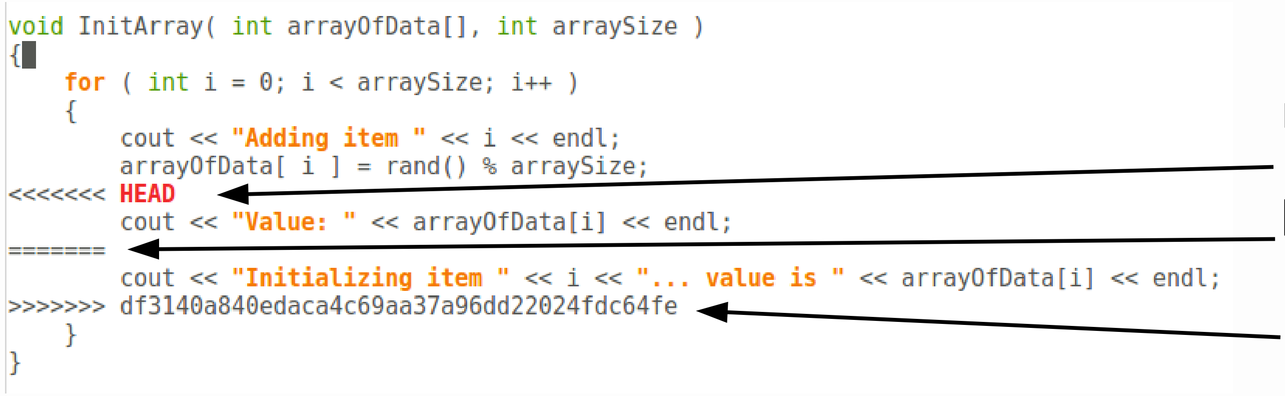
Notes

Source Control:

- Save changes to server
- Keep track of changes over time
- Merge code together
- Share code

2. Features: Merging

And if it can't figure out how to merge (usually if changes are made to the same region of the same file), then it will add markers to the file so you can manually merge as you see fit. This is easier than trying to read two files and figure out where the changes are yourself.



```
void InitArray( int arrayOfData[], int arraySize )
{
    for ( int i = 0; i < arraySize; i++ )
    {
        cout << "Adding item " << i << endl;
        arrayOfData[ i ] = rand() % arraySize;
<<<<<<< HEAD
        cout << "Value: " << arrayOfData[i] << endl;
=======
        cout << "Initializing item " << i << "... value is " << arrayOfData[i] << endl;
>>>>>>> df3140a840edaca4c69aa37a96dd22024fdc64fe
    }
}
```

Then it adds markers to let you know what it couldn't merge on its own.

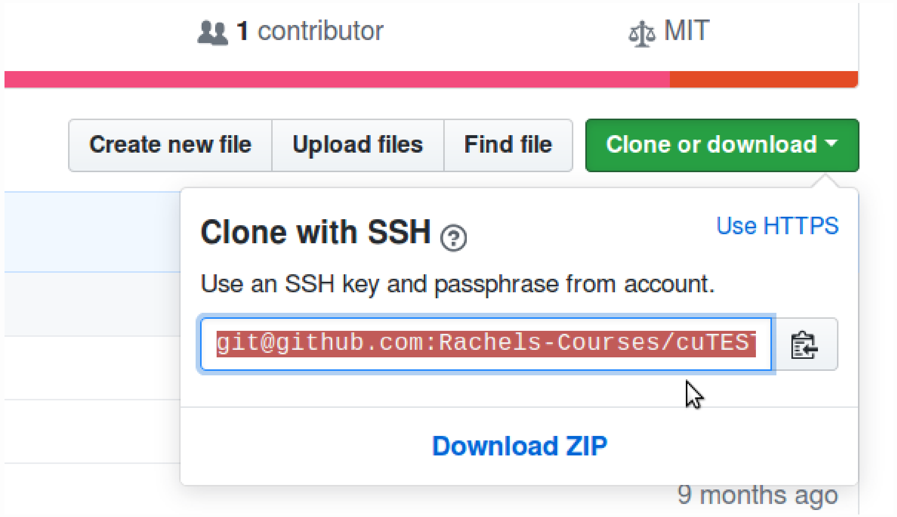
Notes

Source Control:
- Save changes to server
- Keep track of changes over time
- Merge code together
- Share code

2. Features: Sharing

Source Control stores your code on a server, and making a copy of the code on your local machine is as simple as typing a command, like:

```
git clone USER@SERVER:REPOSITORY
```



Use an SSH key and passphrase from account.

```
git@github.com:Rachels-Courses/cuTES
```

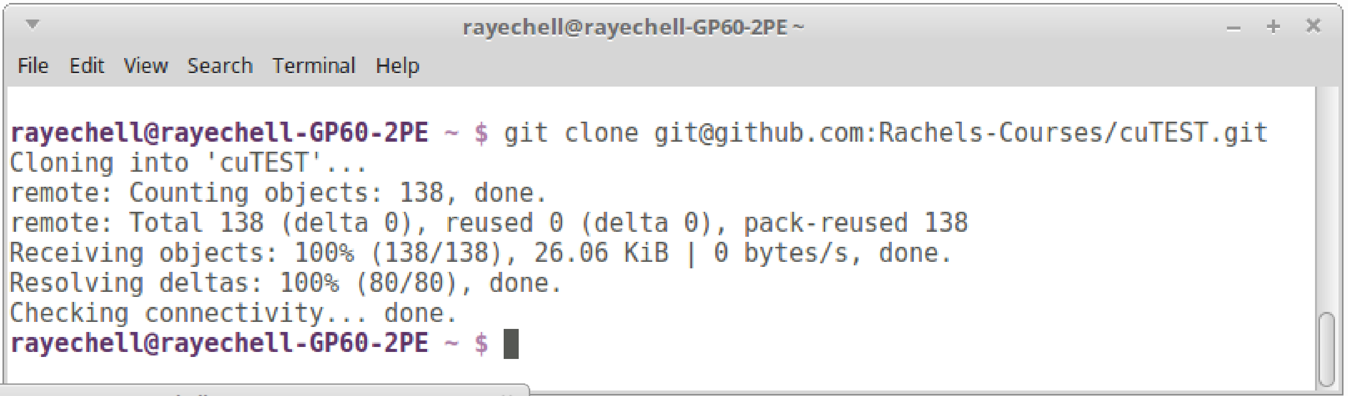
If the server has a web interface, it will show you the URL on the repository's homepage.

Notes

Source Control:

- Save changes to server
- Keep track of changes over time
- Merge code together
- Share code

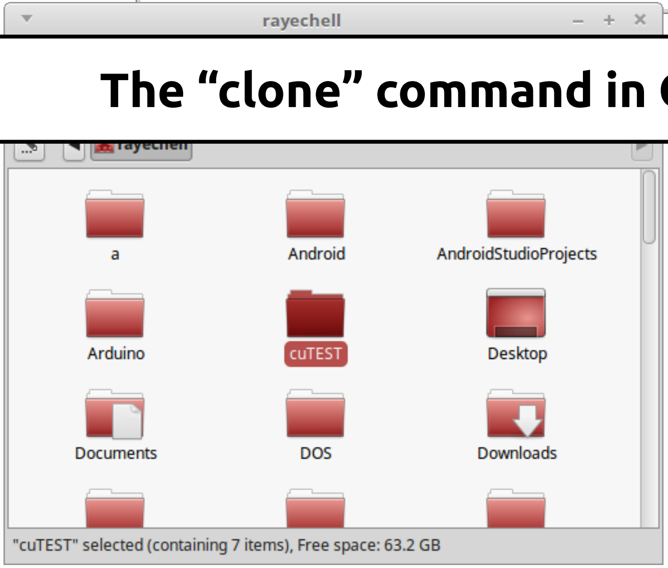
2. Features: Sharing



```
rayechell@rayechell-GP60-2PE ~ $ git clone git@github.com:Rachels-Courses/cuTEST.git
Cloning into 'cuTEST'...
remote: Counting objects: 138, done.
remote: Total 138 (delta 0), reused 0 (delta 0), pack-reused 138
Receiving objects: 100% (138/138), 26.06 KiB | 0 bytes/s, done.
Resolving deltas: 100% (80/80), done.
Checking connectivity... done.
rayechell@rayechell-GP60-2PE ~ $
```

rayechell@rayechell-GP60-2PE ~
File Edit View Search Terminal Help

The “clone” command in Git will pull down all the changes…



"cuTEST" selected (containing 7 items), Free space: 63.2 GB

And all the files will be on your hard drive in a folder with that repository name.

Notes

Source Control:

- Save changes to server
- Keep track of changes over time
- Merge code together
- Share code

2. Features: Server

You can use the Source Control software to create a repository server on your local machine, or on a machine on your network, but there are also online hosting features as well.

If your repository is stored on a server that is internet-accessible, you can pull down your code from anywhere.

Notes

Source Control:

- Save changes to server
- Keep track of changes over time
- Merge code together
- Share code

Source Control Solutions
&
Hosting Options

3. Source Control & Hosting

Some common Source Control solutions are:

- **TFS (Team Foundation Server)**
  Microsoft's Source Control solution, common at businesses where MS tools are used. An alright option.*

- **Git**
  A Source Control solution by the creator of Linux. Common in businesses and open source projects. A good option.

- **Mercurial**
  A Source Control solution by Atlassian, who also run Jira and Confluence (other development tools). Also a good option.

- **SVN (Subversion)**
  An older Source Control system. Still sometimes used, but not a great option.

** Bias opinion; just assume I'm not in love with any Microsoft products and prefer Linux and Open Source / Free Software.*

Notes

Source Control:
- Save changes to server
- Keep track of changes over time
- Merge code together
- Share code

3. Source Control & Hosting

Once you're using a Source Control solution, you need a way to host your repositories. There are some services online, like:

- **CodePlex** **codeplex.com**
  Hosting for Open Source projects ran by Microsoft. Usually stores .NET projects. Supports TFS, Git, Mercurial, and SVN.

- **GitHub** **github.com**
  Hosting for Git-based projects.

- **BitBucket** **bitbucket.org**
  Ran by Atlassian, supports Mercurial and Git.

- **SourceForge** **sourceforge.net**
  A popular place for (older?) open source projects. Supports SVN, Git, and Mercurial.

Notes

Source Control:

- Save changes to server
- Keep track of changes over time
- Merge code together
- Share code

3. Source Control & Hosting

Most of these solutions are free for open source projects, where hosting private repositories may cost money.

The exception here is BitBucket, which allows you to have unlimited private repositories, so long as your team size is 5 people or fewer.

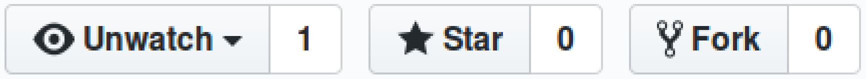
Notes

Source Control:
- Save changes to server
- Keep track of changes over time
- Merge code together
- Share code

3. Source Control & Hosting

I use GitHub for my Open Source stuff, and BitBucket for my private repositories.

GitHub is better equipped for "social coding" and following peoples' projects.



You can also find projects like **Linux**, **DOOM**, **Prince of Persia**, and other notable open source projects here.

Notes

Source Control:

- Save changes to server
- Keep track of changes over time
- Merge code together
- Share code

Getting Started with BitBucket

4. Getting Started with BitBucket

For this class, you will be using Git BitBucket to keep track of your code.

The setup process for BitBucket is similar to it is in GitHub, so you can easily move between each in the future if you'd like.

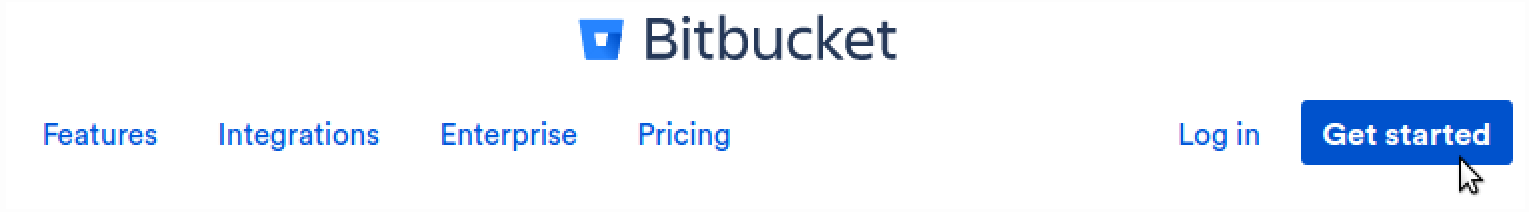
Notes

Source Control:

- Save changes to server
- Keep track of changes over time
- Merge code together
- Share code

4. Getting Started with BitBucket

On the front page of **bitbucket.org**, click “Get started”.

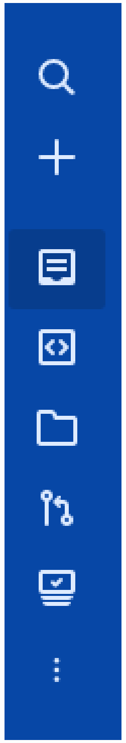


Go through the account registration process…

Notes

4. Getting Started with BitBucket

From the BitBucket dashboard, there are a series of buttons on the left-hand side:



Search (for repository, code, etc.)

Create a new repository

Overview (dashboard)

Repositories

Projects

Pull requests

Issues

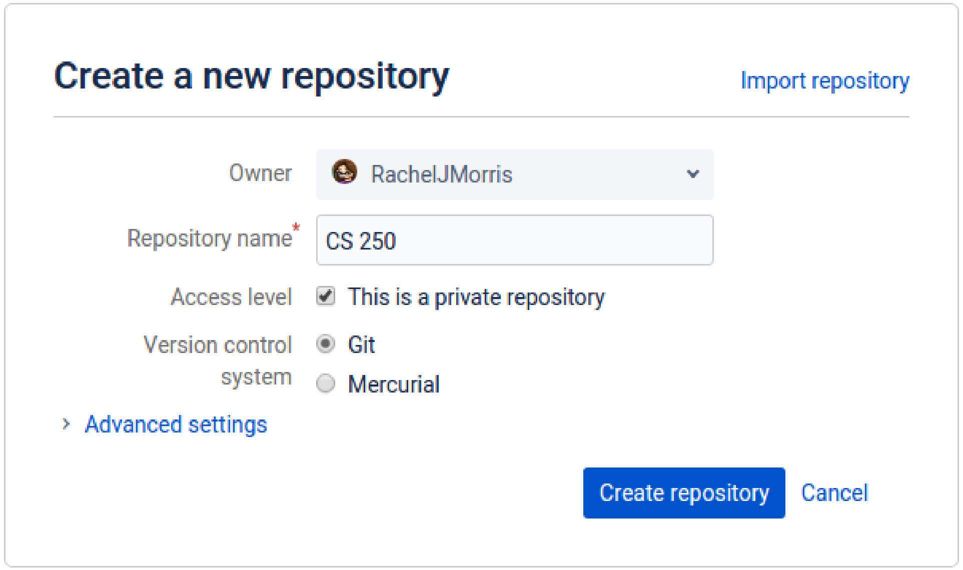
More

Notes

4. Getting Started with BitBucket

Select “Create a new repository” (the + button).

Give your repository a name, make sure it’s **private**, select **Git** as the version control system, and click “Create repository”.



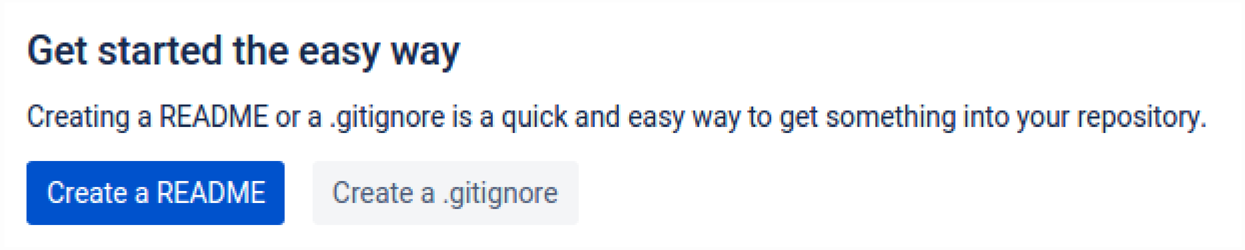
Create a new repository

Owner	RachelJMorris
Repository name*	CS 250
Access level	[✓] This is a private repository
Version control system	[● Git] ○ Mercurial

Notes

4. Getting Started with BitBucket

Once created, add the first file into your repository by clicking the "Create a README" button.

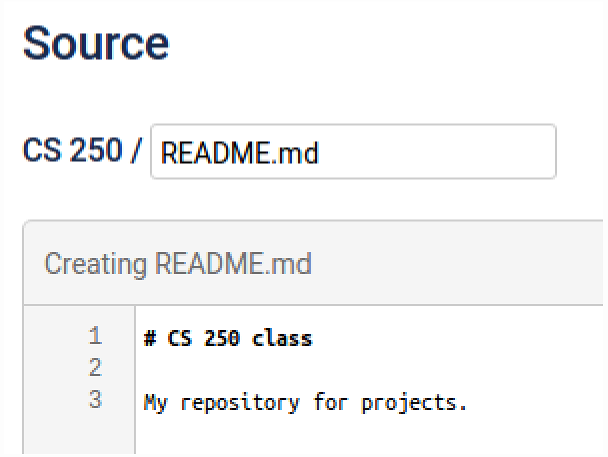


Get started the easy way

Creating a README or a .gitignore is a quick and easy way to get something into your repository.

BitBucket (as well as GitHub) has a web-based text editor that you can use to modify code or text files. Go ahead and erase the default stuff in the Readme and add some info, then click "Commit".

On the Commit changes screen, click "Commit".



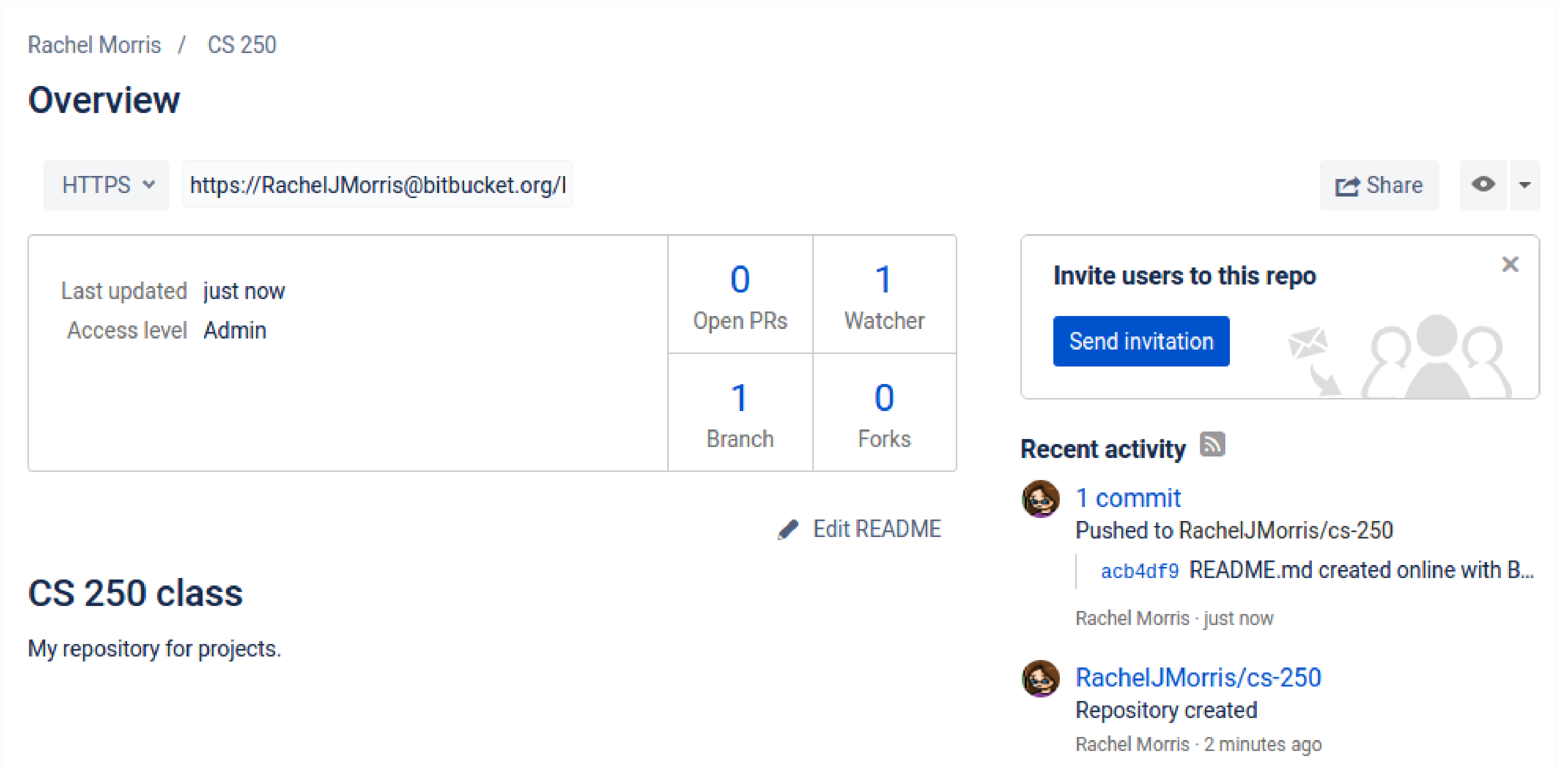
```
# CS 250 class

My repository for projects.
```

Notes

4. Getting Started with BitBucket

Your repository homepage will look like this now.



Rachel Morris / CS 250

Overview

HTTPS https://RachelJMorris@bitbucket.org/l

Share

Last updated just now Access level Admin	0 Open PRs	1 Watcher
	1 Branch	0 Forks

Invite users to this repo

Send invitation

Edit README

CS 250 class

My repository for projects.

Recent activity

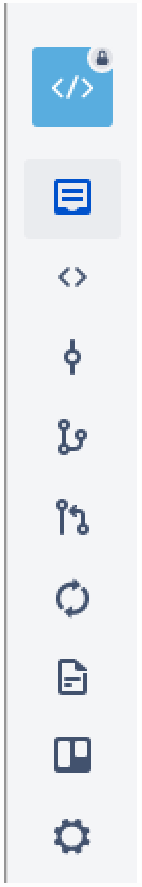
1 commit
Pushed to RachelJMorris/cs-250
acb4df9 README.md created online with B...
Rachel Morris · just now

RachelJMorris/cs-250
Repository created
Rachel Morris · 2 minutes ago

Notes

4. Getting Started with BitBucket

On the left side is a toolbar of buttons for your repo.



Repository homepage

Overview

Source (view all files in the web interface)

Commits (view a list of all changes)

Branches

Pull requests

Pipelines

Downloads

Boards

Settings

Notes

4. Getting Started with BitBucket

Next we need to work with Git before much of anything shows up in the web interface…

Notes

Getting Started with Git

5. Getting Started with Git

Git is a software program you will need to download. The lab machines have Git already, but if you'll be working from your personal machine you will need to download it.

Download Git from **git-scm.com** if you're in Windows (or Mac?)

If you're in Linux, you can install it via your package manager.

Notes

5. Getting Started with Git

Git has a graphical (GUI) client, but we will be using the command line interface (CLI) for simplicity.

In Windows, you can access it via “Git Bash”, and in Linux (and Mac?) you can access Git simply from the Terminal.

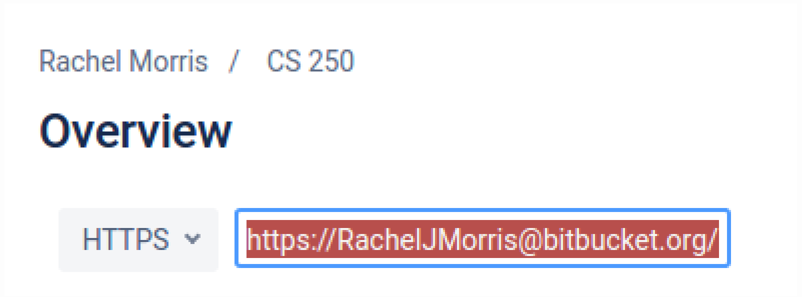
Notes

5. Getting Started with Git

First, we will need to bring down the repository to your local machine. This is called **cloning**.

```
git clone URL
```

To get the URL, go to your BitBucket page and copy this URL:


Rachel Morris / CS 250

Overview

HTTPS https://RachelJMorris@bitbucket.org/

Notes

git clone URL
Make a copy of the repository

5. Getting Started with Git

Locate a place on your hard drive where you want to keep your projects (or on the school computer just use the Desktop to keep it easy)

If you're on Windows, right-click empty space in the folder and click "Open in Git Bash".

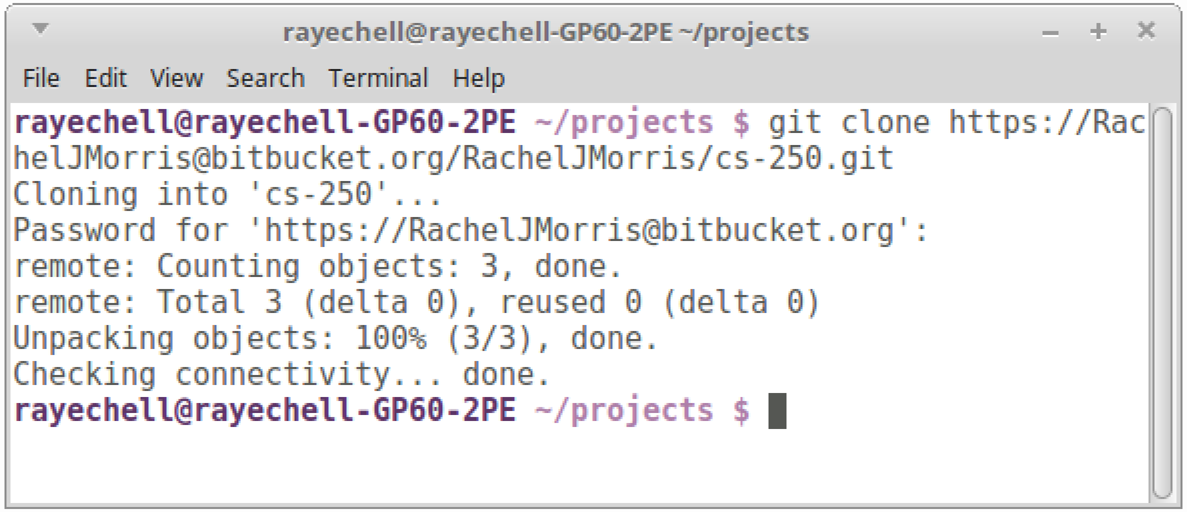
Otherwise, open the Terminal here in Linux/Mac.

Notes

git clone URL
Make a copy of the repository

5. Getting Started with Git

Type in **git clone**, then paste in the URL for your repository.


rayechell@rayechell-GP60-2PE ~/projects
File Edit View Search Terminal Help

```
rayechell@rayechell-GP60-2PE ~/projects $ git clone https://Rac
helJMorris@bitbucket.org/RachelJMorris/cs-250.git
Cloning into 'cs-250'...
Password for 'https://RachelJMorris@bitbucket.org':
remote: Counting objects: 3, done.
remote: Total 3 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (3/3), done.
Checking connectivity... done.
rayechell@rayechell-GP60-2PE ~/projects $
```

Now your project folder will be available in the directory where you cloned the repository.

Notes

git clone URL
Make a copy of the repository

5. Getting Started with Git

You will want to set up some config info if this is your first time using Git on this computer.

Enter the following command:

```
git config --global user.name "YOUR NAME"
```

And then:

```
git config --global user.email YOUREMAIL@EXAMPLE.COM
```

Close the terminal once you're done.

Notes

git clone URL
Make a copy of the repository

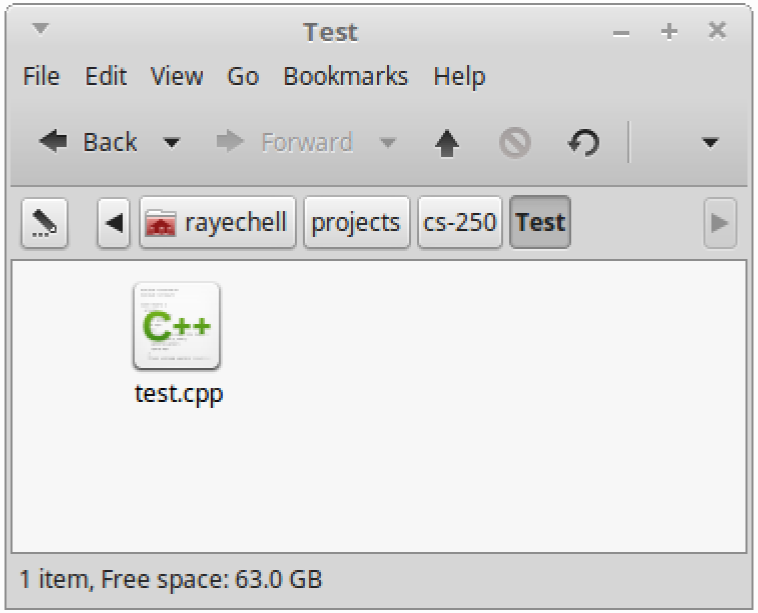
5. Getting Started with Git

Within your project folder, create a new folder called “Test”.

Within that folder, create a source file named “test.cpp”.

Just paste in some simple code like this:

```
int main()
{
    return 0;
}
```



1 item, Free space: 63.0 GB

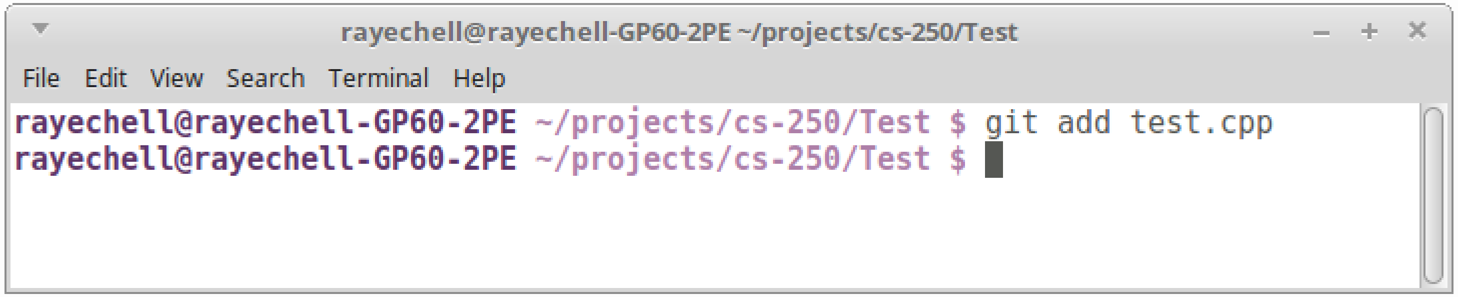
Notes

git clone URL
Make a copy of the repository

5. Getting Started with Git

Open the terminal in the folder you're currently in. Add this new file to a changeset using

`git add FILENAME`


rayechell@rayechell-GP60-2PE ~/projects/cs-250/Test
File Edit View Search Terminal Help

```
rayechell@rayechell-GP60-2PE ~/projects/cs-250/Test $ git add test.cpp
rayechell@rayechell-GP60-2PE ~/projects/cs-250/Test $
```

Notes

git clone URL
Make a copy of the repository

git add FILE
Add a file to a changeset

5. Getting Started with Git

You will have to add every file that you want to push to the server, but you can use some shortcuts…

Add all cpp files in this directory and subdirectories:

```
git add *.cpp
```

Add ALL FILES in this directory and subdirectories:

```
git add .
```

You may not want to add ALL files in the project directory to your repository. For example, when you compile your project your compiler will generate temporary files, so it's better to add all ".cpp files" and all ".hpp files" instead of add "all files".

Notes

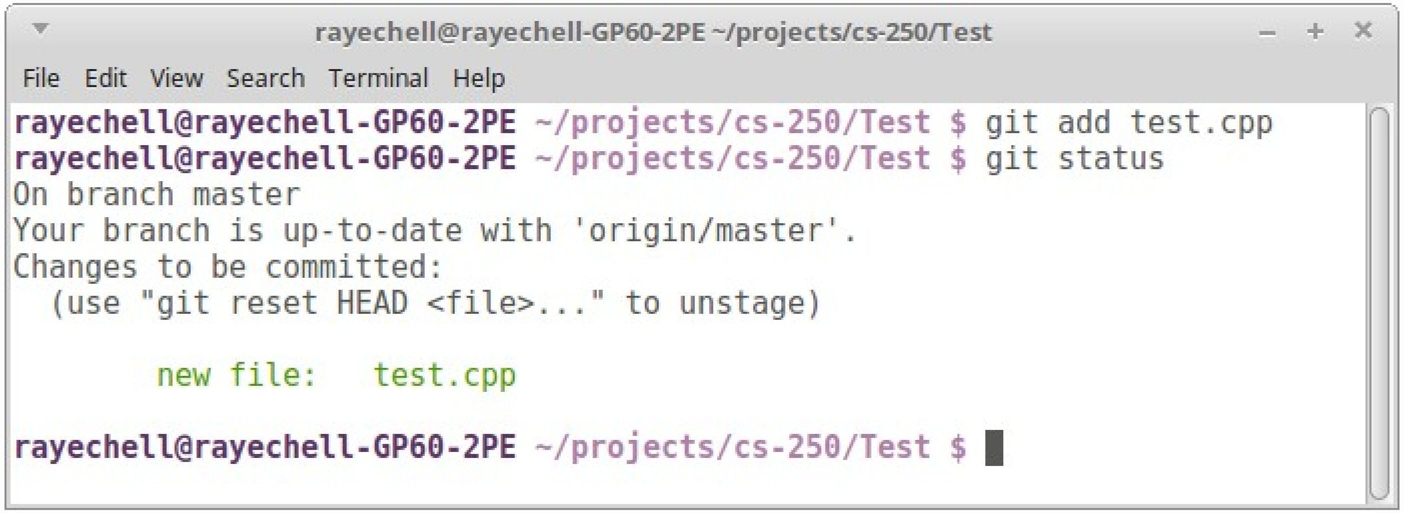
git clone URL
Make a copy of the repository

git add FILE
Add a file to a changeset

5. Getting Started with Git

To view the changes you have ready to go, use

`git status`


rayechell@rayechell-GP60-2PE ~/projects/cs-250/Test
File Edit View Search Terminal Help

```
rayechell@rayechell-GP60-2PE ~/projects/cs-250/Test $ git add test.cpp
rayechell@rayechell-GP60-2PE ~/projects/cs-250/Test $ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   test.cpp

rayechell@rayechell-GP60-2PE ~/projects/cs-250/Test $ 
```

Notes

git clone URL
Make a copy of the repository

git add FILE
Add a file to a changeset

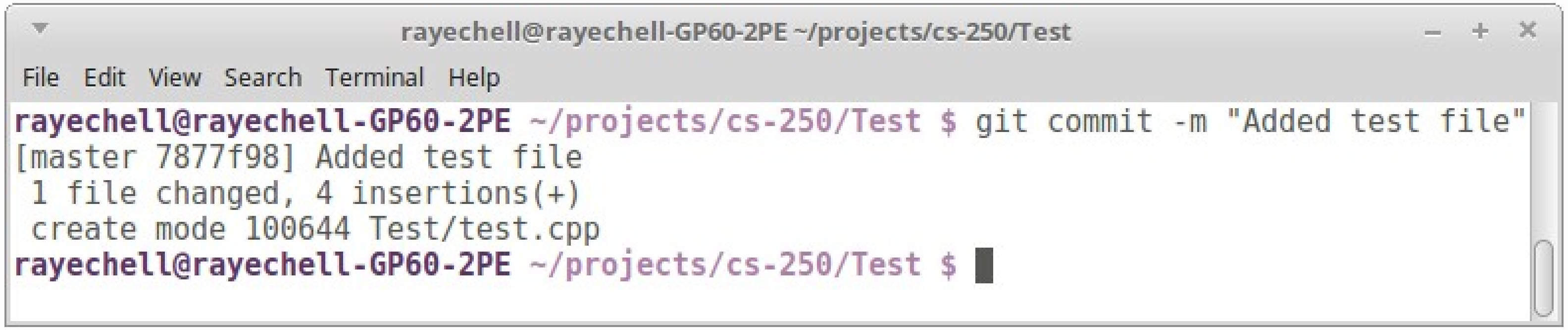
git status
View changes ready to be committed

5. Getting Started with Git

Once you have all the files you've changed ready to go, you will make a **commit**. A commit makes a snapshot in time of all your files. Commit your changes with:

```
git commit -m “message”
```

Your message should describe what you changed.


rayechell@rayechell-GP60-2PE ~/projects/cs-250/Test
File Edit View Search Terminal Help

```
rayechell@rayechell-GP60-2PE ~/projects/cs-250/Test $ git commit -m "Added test file"
[master 7877f98] Added test file
 1 file changed, 4 insertions(+)
 create mode 100644 Test/test.cpp
rayechell@rayechell-GP60-2PE ~/projects/cs-250/Test $
```

Notes

git clone URL
Make a copy of the repository

git add FILE
Add a file to a changeset

git status
View changes ready to be committed

git commit -m “notes”
Make a snapshot of your changes

5. Getting Started with Git

Even though you've committed your changes, they won't be on the server yet – they've only been saved on your local machine. You would use a push command to push your changes to the server.

However, if you've been working with another person, or on multiple machines, you might want to do a **pull** before you do a **push**.

Notes

git clone URL
Make a copy of the repository

git add FILE
Add a file to a changeset

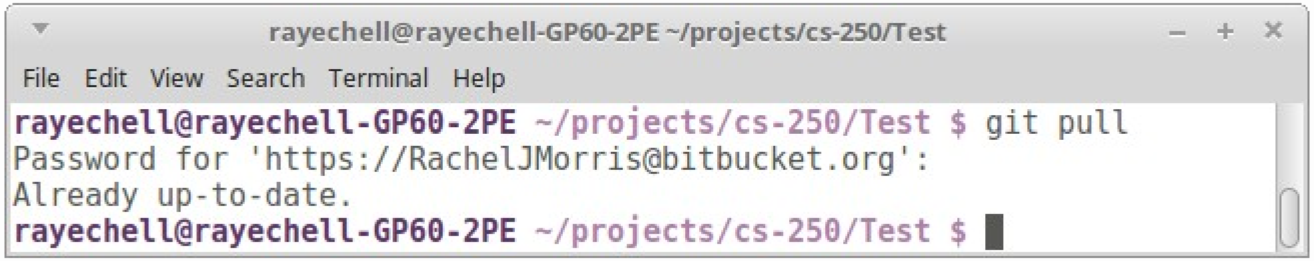
git status
View changes ready to be committed

git commit -m "notes"
Make a snapshot of your changes

5. Getting Started with Git

To pull changes from the server, use:

git pull


rayechell@rayechell-GP60-2PE ~/projects/cs-250/Test
File Edit View Search Terminal Help

```
rayechell@rayechell-GP60-2PE ~/projects/cs-250/Test $ git pull
Password for 'https://RachelJMorris@bitbucket.org':
Already up-to-date.
rayechell@rayechell-GP60-2PE ~/projects/cs-250/Test $
```

In this case it will probably just say “Already up-to-date”, unless you’ve made changes from the web interface.

Notes

git clone URL
Make a copy of the repository

git add FILE
Add a file to a changeset

git status
View changes ready to be committed

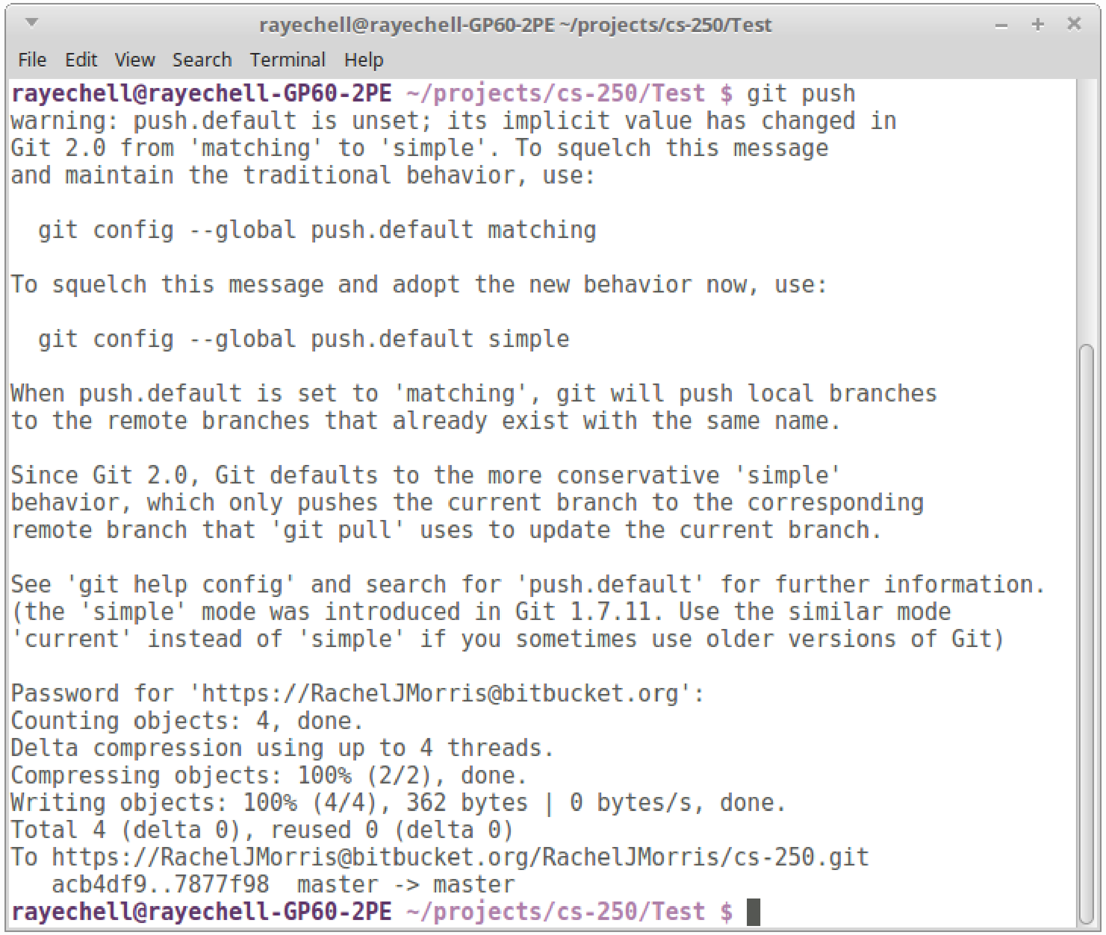
git commit -m “notes”
Make a snapshot of your changes

git pull
Pull changes from the server

5. Getting Started with Git

To push your changes to the server, use:

`git push`


rayechell@rayechell-GP60-2PE ~/projects/cs-250/Test
File Edit View Search Terminal Help

```
rayechell@rayechell-GP60-2PE ~/projects/cs-250/Test $ git push
warning: push.default is unset; its implicit value has changed in
Git 2.0 from 'matching' to 'simple'. To squelch this message
and maintain the traditional behavior, use:

  git config --global push.default matching

To squelch this message and adopt the new behavior now, use:

  git config --global push.default simple

When push.default is set to 'matching', git will push local branches
to the remote branches that already exist with the same name.

Since Git 2.0, Git defaults to the more conservative 'simple'
behavior, which only pushes the current branch to the corresponding
remote branch that 'git pull' uses to update the current branch.

See 'git help config' and search for 'push.default' for further information.
(the 'simple' mode was introduced in Git 1.7.11. Use the similar mode
'current' instead of 'simple' if you sometimes use older versions of Git)

Password for 'https://RachelJMorris@bitbucket.org':
Counting objects: 4, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 362 bytes | 0 bytes/s, done.
Total 4 (delta 0), reused 0 (delta 0)
To https://RachelJMorris@bitbucket.org/RachelJMorris/cs-250.git
   acb4df9..7877f98  master -> master
rayechell@rayechell-GP60-2PE ~/projects/cs-250/Test $
```

Notes

git clone URL
Make a copy of the repository

git add FILE
Add a file to a changeset

git status
View changes ready to be committed

git commit -m “notes”
Make a snapshot of your changes

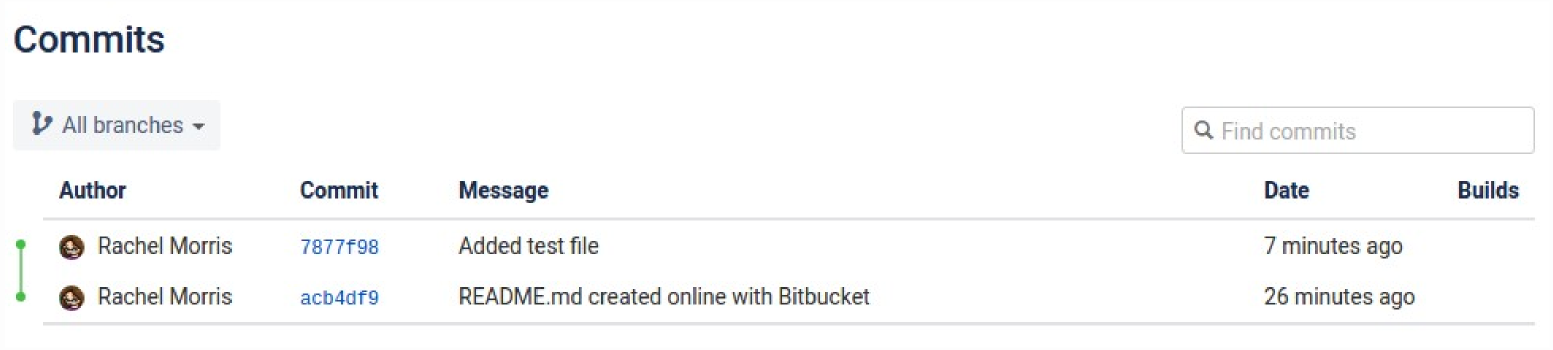
git pull
Pull changes from the server

git push
Push changes to server

5. Getting Started with Git

Now that you have some changes pushed up, go back to the web interface of your repository and refresh.

Click on the commits button to view your commits



Commits

Author	Commit	Message	Date	Builds
Rachel Morris	7877f98	Added test file	7 minutes ago	
Rachel Morris	acb4df9	README.md created online with Bitbucket	26 minutes ago	

Notes

git clone URL
Make a copy of the repository

git add FILE
Add a file to a changeset

git status
View changes ready to be committed

git commit -m "notes"
Make a snapshot of your changes

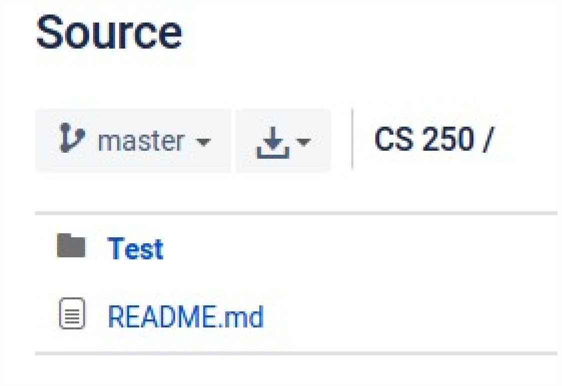
git pull
Pull changes from the server

git push
Push changes to server

5. Getting Started with Git

Now that you have some changes pushed up, go back to the web interface of your repository and refresh.

Click on the source button <> to view your source from the web interface.



Notes

git clone URL
Make a copy of the repository

git add FILE
Add a file to a changeset

git status
View changes ready to be committed

git commit -m “notes”
Make a snapshot of your changes

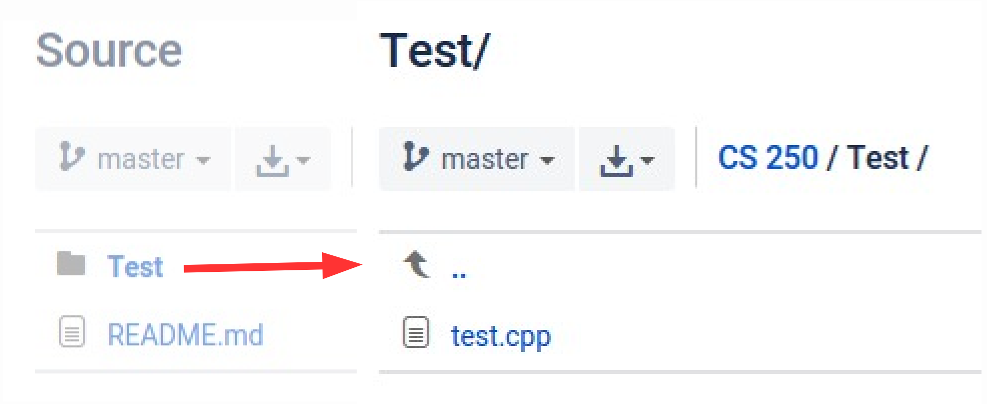
git pull
Pull changes from the server

git push
Push changes to server

5. Getting Started with Git

Now that you have some changes pushed up, go back to the web interface of your repository and refresh.

Click on the source button ‹› to view your source from the web interface.



Notes

git clone URL
Make a copy of the repository

git add FILE
Add a file to a changeset

git status
View changes ready to be committed

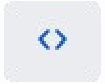
git commit -m “notes”
Make a snapshot of your changes

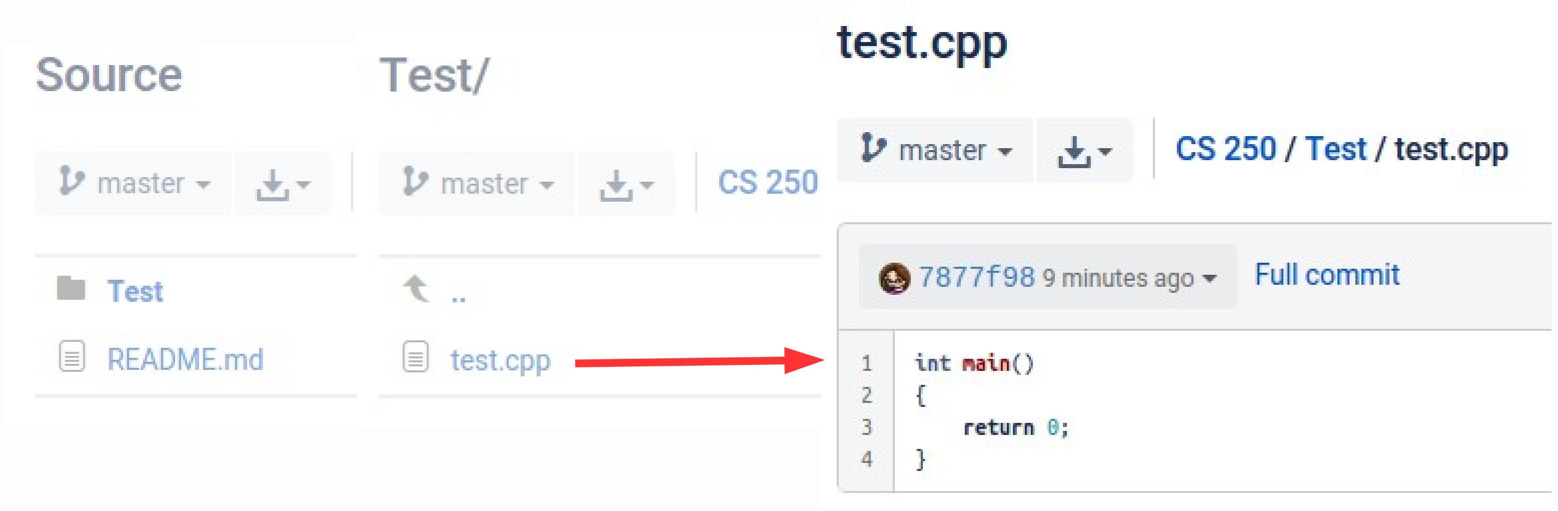
git pull
Pull changes from the server

git push
Push changes to server

5. Getting Started with Git

Now that you have some changes pushed up, go back to the web interface of your repository and refresh.

Click on the source button  to view your source from the web interface.



```
int main()
{
    return 0;
}
```

Notes

git clone URL
Make a copy of the repository

git add FILE
Add a file to a changeset

git status
View changes ready to be committed

git commit -m "notes"
Make a snapshot of your changes

git pull
Pull changes from the server

git push
Push changes to server

Conclusion

We scratched the surface of using Source Control and Git.

Source Control is an important tool for any programmer, and it will also save you a ton of headaches as you’re programming.

Make sure to keep your projects up-to-date in your repository, as you will have to share it at the end of the semester for credit.