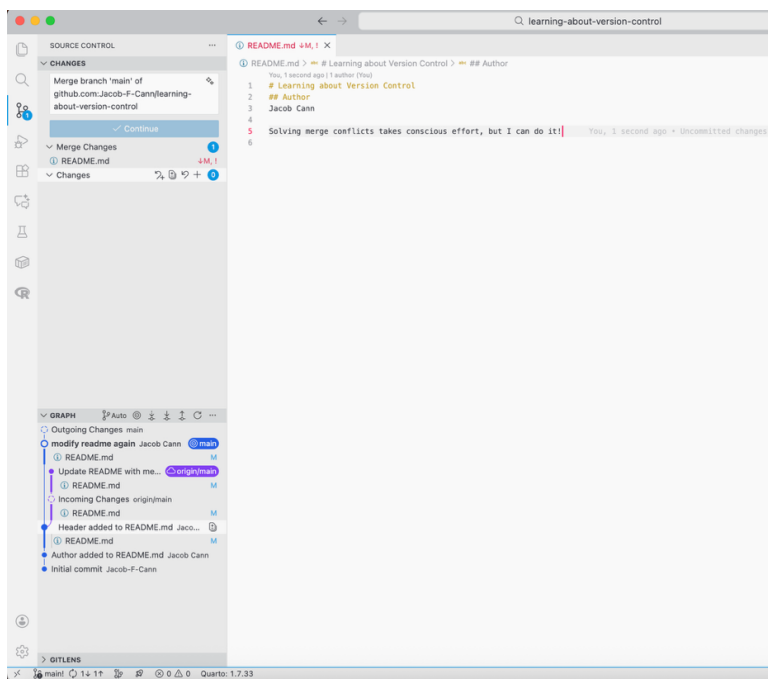
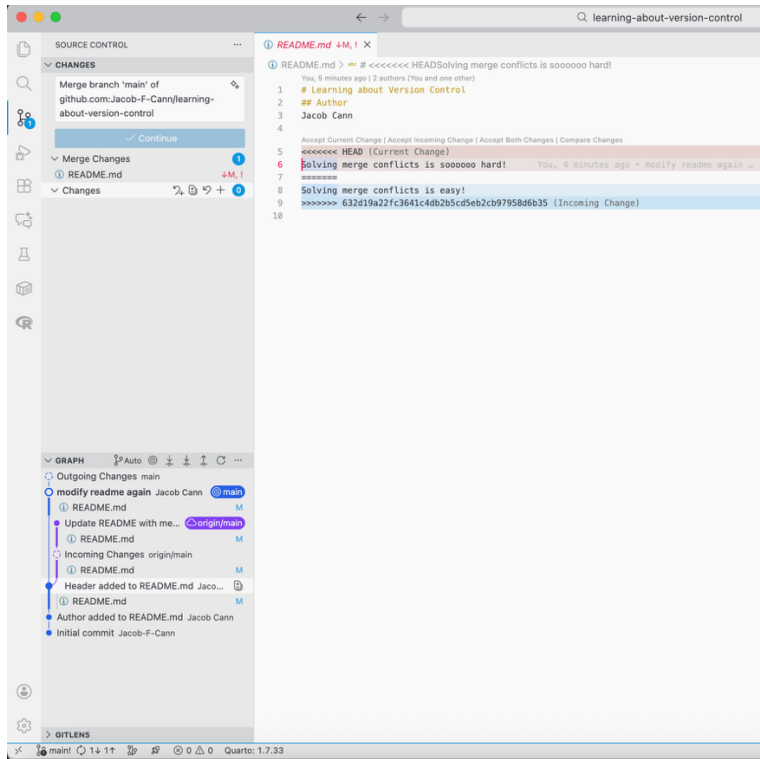


DSCI 552 – Individual Assignment 1

1. Link to Commits

<https://github.com/Jacob-F-Cann/learning-about-version-control/commits/main/>

2. Screenshots



```

(base) jacob@dhcp-128-189-253-147 ~/Personal GitHub/learning-about-version-control (main)
$ open README.md

(base) jacob@dhcp-128-189-253-147 ~/Personal GitHub/learning-about-version-control (main)
$ git add .

(base) jacob@dhcp-128-189-253-147 ~/Personal GitHub/learning-about-version-control (main +)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   README.md

(base) jacob@dhcp-128-189-253-147 ~/Personal GitHub/learning-about-version-control (main +)
$ git commit -m "modify readme again"
[main 1468b4d] modify readme again
1 file changed, 3 insertions(+), 1 deletion(-)

(base) jacob@dhcp-128-189-253-147 ~/Personal GitHub/learning-about-version-control (main)
$ git push
To github.com:Jacob-F-Cann/learning-about-version-control.git
 ! [rejected]        main -> main (fetch first)
error: failed to push some refs to 'github.com:Jacob-F-Cann/learning-about-version-control.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.

(base) jacob@dhcp-128-189-253-147 ~/Personal GitHub/learning-about-version-control (main)
$ git pull
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 1007 bytes | 503.00 KiB/s, done.
From github.com:Jacob-F-Cann/learning-about-version-control
   d507261..632d19a  main       -> origin/main
hint: You have divergent branches and need to specify how to reconcile them.
hint: You can do so by running one of the following commands sometime before
hint: your next pull:
hint:
hint:   git config pull.rebase false  # merge
hint:   git config pull.rebase true   # rebase
hint:   git config pull.ff only        # fast-forward only
hint:
hint: You can replace "git config" with "git config --global" to set a default
hint: preference for all repositories. You can also pass --rebase, --no-rebase,
hint: or --ff-only on the command line to override the configured default per
hint: invocation.
fatal: Need to specify how to reconcile divergent branches.

(base) jacob@dhcp-128-189-253-147 ~/Personal GitHub/learning-about-version-control (main)
$ git pull --rebase=false
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
Automatic merge failed; fix conflicts and then commit the result.

(base) jacob@dhcp-128-189-253-147 ~/Personal GitHub/learning-about-version-control (main +[MERGING])
$ git add .

(base) jacob@dhcp-128-189-253-147 ~/Personal GitHub/learning-about-version-control (main +[MERGING])
$ git commit -m "Readme merge fix"
[main 7249fbb] Readme merge fix

(base) jacob@dhcp-128-189-253-147 ~/Personal GitHub/learning-about-version-control (main)
$ git push
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 10 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 629 bytes | 629.00 KiB/s, done.
Total 6 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:Jacob-F-Cann/learning-about-version-control.git
   632d19a..7249fbb  main -> main

(base) jacob@dhcp-128-189-253-147 ~/Personal GitHub/learning-about-version-control (main)
$

```

3. Branch-pull-request Workflow Explanation:

The main advantage of using branches and pull requests when instead of pushing directly to the main branch is avoiding frequent merge conflicts and avoiding breaking the current working codebase. You can continue working on individual modular pieces of your collaborative project by isolating your changes to your specific branch until you are ready to send a pull request, to allow for review of your changes. This makes merge conflict resolution more robust through peer review and saves a lot of headache by making merges more intentional.

README.md File:

Learning about Version Control

Author

Jacob Cann

Solving merge conflicts takes conscious effort, but I can do it!

I think version control is great for tracking your changes and making sure your changes are backed up in case anything happens to your local hard drive.

The biggest source of confusion for me so far is merge conflicts and resolving them the simplest way possible. For straightforward one-line examples it is

easily understandable but I can imagine how confusing it would get for complicated conflicts across many lines of code.