

Git / GitHub Workshop

How To HackRPI

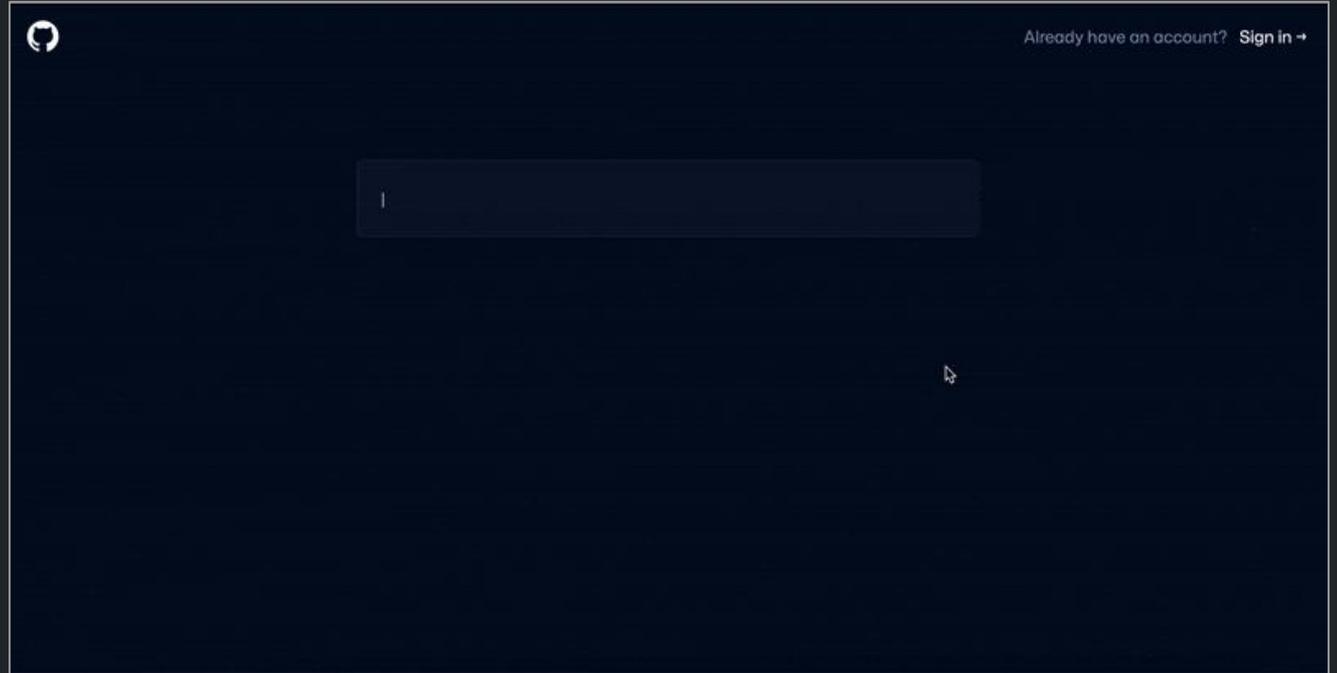


How to Set Up *GitHub*

Make a GitHub
account

Use your
personal email!

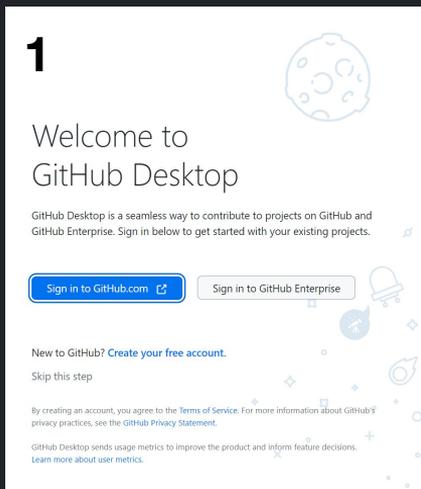
<https://github.com/>



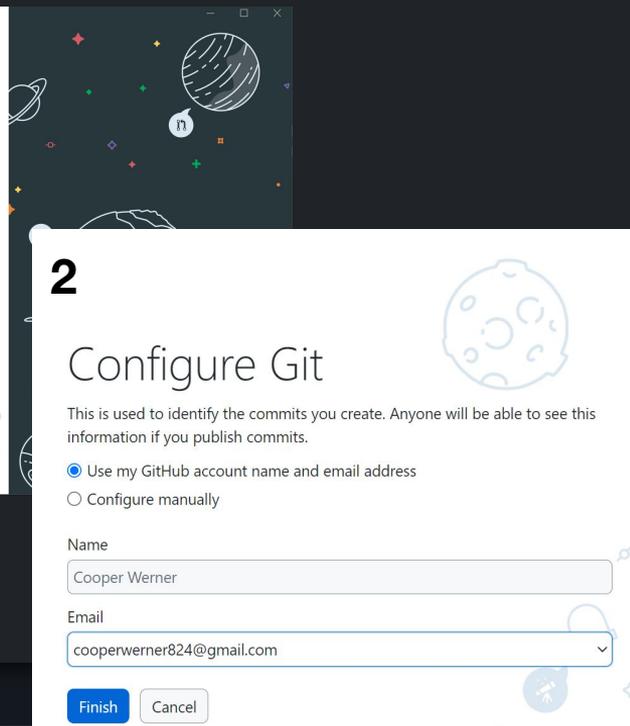
How to Install *GitHub Desktop*

A client to interact with GitHub
from our computer.

<https://desktop.github.com>



^Download the installer from
the website and use the
installer to install the software.



Introduction

H **A** **C** **K**



What are Git and GitHub?



git

- **Git**
 - Track changes to the codebase
 - Allows you to easily reverse changes
 - Improves coordination between developers
- **GitHub**
 - Storing our code on the web
 - Enables many Devs on one project
 - Makes deploying code easier



How do we use Git and GitHub

- A variety of commands and concepts
 - Commits
 - Branches
 - Pushes / Pulls
 - Forks
 - Pull Requests / Merges
- We will go more in depth with these later but let's get setup first.



Using Git / GitHub

H **A** **C** **K**



Create a Repository

Repositories are a place to store code for a project

1. Go to <https://github.com>
2. Click the green “New” button.
3. And follow the prompts to create a new repository.
4. Make sure Repository is public

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

Repository template

No template

Start your repository with a template repository's contents.

Owner *

3791xk

Repository name *

ds-workshop

ds-workshop is available.

Great repository names are short and memorable. Need inspiration? How about [expert-engine](#)?

Description (optional)

Learning GitHub - DS Lecture Takeover

Public

Anyone on the internet can see this repository. You choose who can commit.

Private

You choose who can see and commit to this repository.

Initialize this repository with:

Add a README file

This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

.gitignore template: C++

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

License: MIT License

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

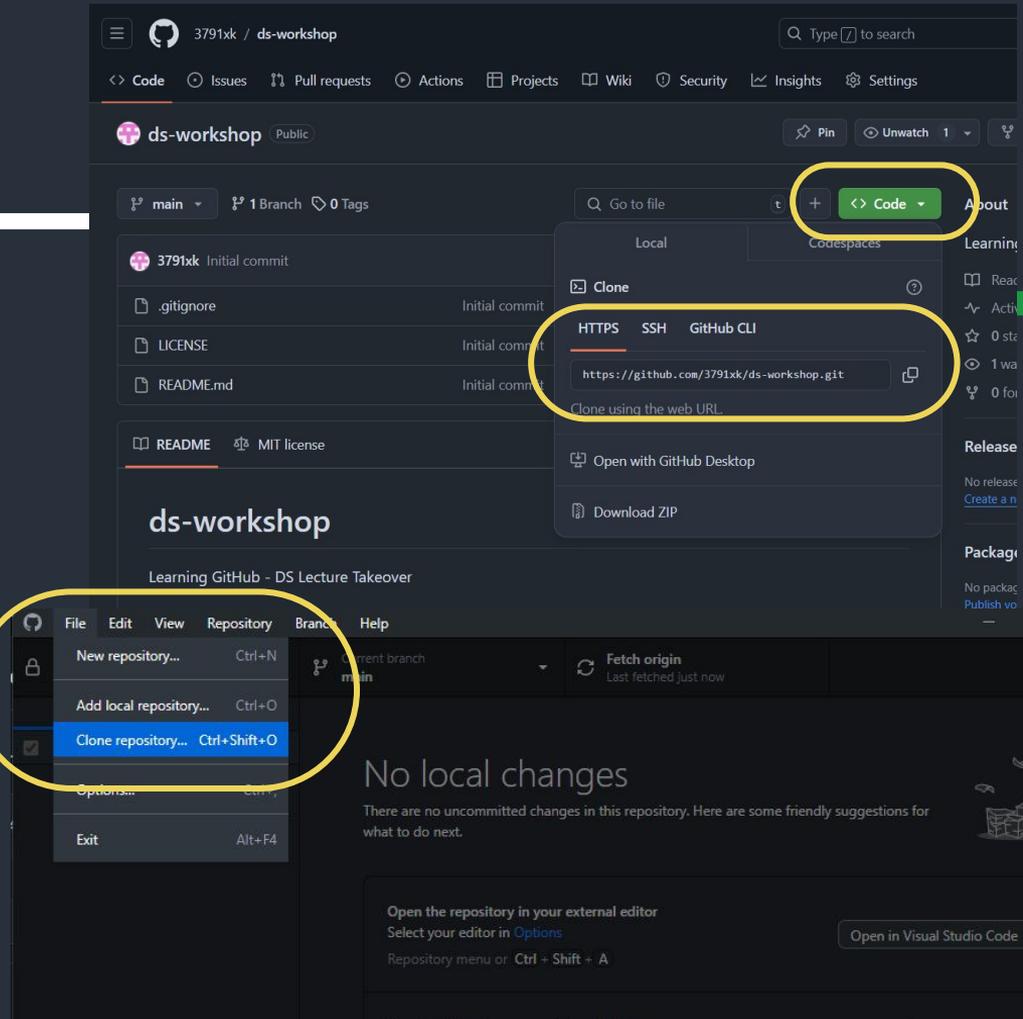
This will set `main` as the default branch. Change the default name in your [settings](#).

You are creating a public repository in your personal account.

Create repository

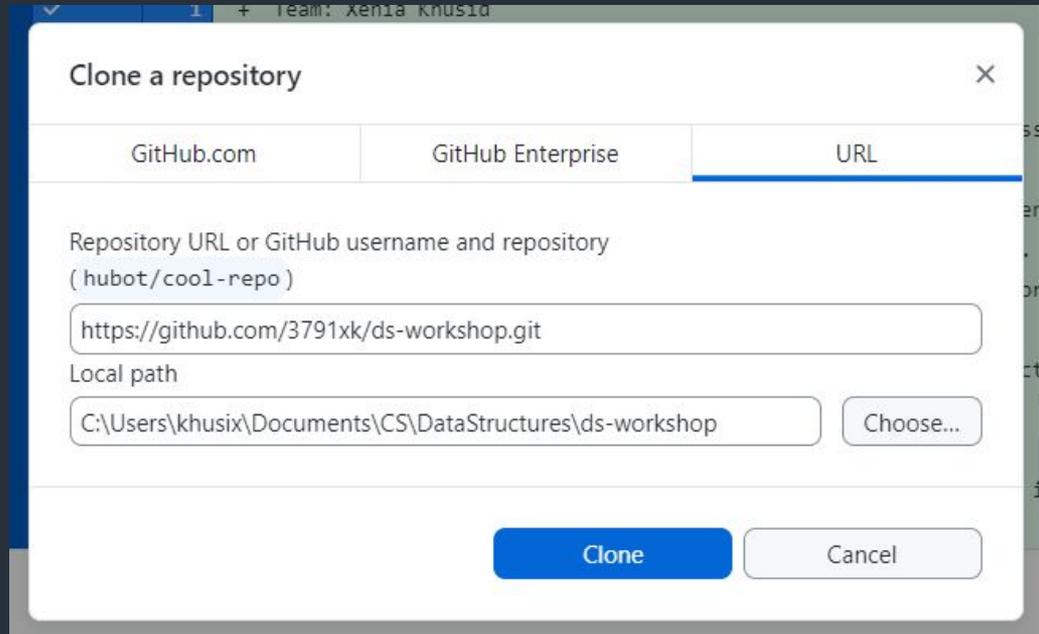
Clone a Repository

1. Creating a copy of the repository on our computer
2. On github.com click the green code button in your repository, copy the link to your repository.
3. Open GitHub Desktop and click file, then Clone repository.



Clone a repository continued:

1. Paste the link from your github repository into the Repository URL field
2. Click the blue clone button



Start our code



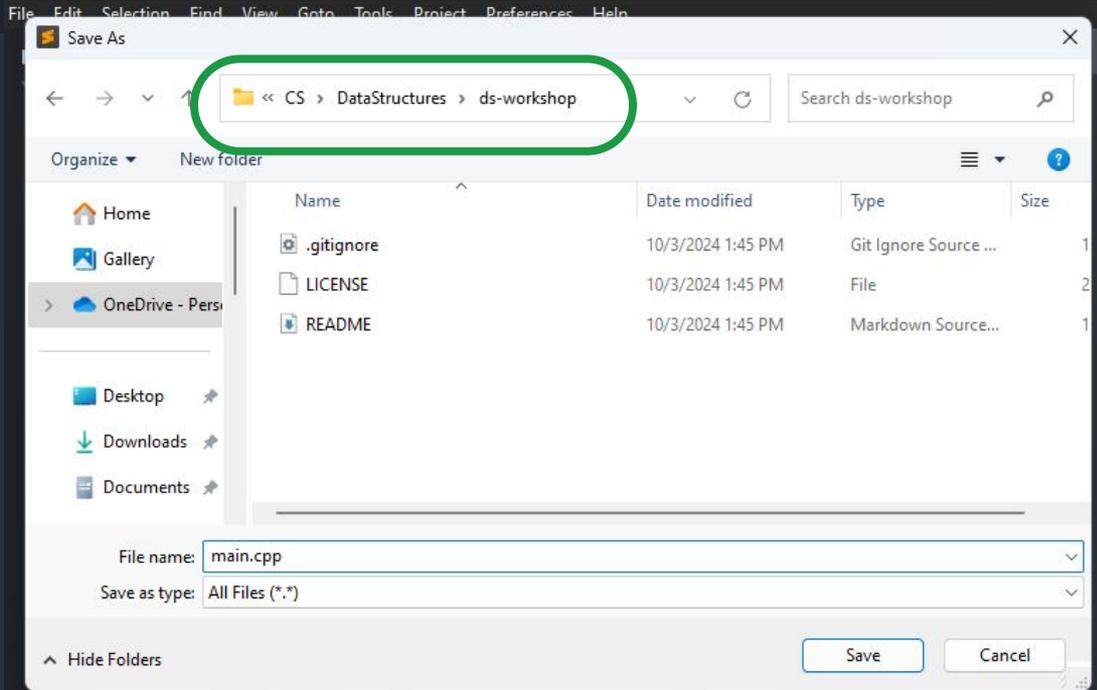
Make a .cpp file in that folder



Find View Goto Tools Project Preferences Help

main.cpp

1



Write a program:

- Accept command line arguments
- Put each object into a `std::list` of `int` values
- Use an iterator to print each value in the list



```
khusix@LAPTOP-NGACK2CL: ~$ g++ main.cpp -g -Wall -Wextra
khusix@LAPTOP-NGACK2CL: /mnt/c/Users/khusix/Documents/CS/DataStructures/ds-workshop$ ./a.out 1 2 3 5 8
1
2
3
5
8
khusix@LAPTOP-NGACK2CL: /mnt/c/Users/khusix/Documents/CS/DataStructures/ds-workshop$
```

<https://github.com/3791xk/ds-workshop>

Example:

Will look something like this:

Or copy from my repo!

[https://github.com/3791xk/
ds-workshop](https://github.com/3791xk/ds-workshop)

```
main.cpp
1  #include <iostream>
2  #include <list>
3  #include <iterator>
4  #include <cstdlib>
5
6  int main(int argc, char* argv[]) {
7
8      std::list<int> intList;
9
10     for (int i = 1; i < argc; ++i) {
11         int temp = std::atoi(argv[i]);
12         intList.push_back(temp);
13     }
14
15     std::list<int>::iterator it = intList.begin();
16     while (it != intList.end()) {
17         std::cout << *it << std::endl;
18         it++;
19     }
20
21     return 0;
22 }
```



See your changes in GitHub Desktop

- Notice “Current branch” is “main”
- One changed file is the one you made and edited
- Write a **name** and **description**
- This is a “**staged commit**”

The screenshot displays the GitHub Desktop interface. At the top, the menu bar includes File, Edit, View, Repository, Branch, and Help. Below the menu, the 'Current repository' is 'ds-workshop' and the 'Current branch' is 'main'. A 'Fetch origin' button indicates the last fetch was 12 minutes ago. The 'Changes' tab shows '1 changed file' and 'main.cpp'. The code editor displays the content of main.cpp, which is a C++ program using a list and an iterator. A commit dialog box is open, with the title 'Started basic list program' and the description 'Take input from command line arguments and print out the same list with an iterator'. The 'Commit to main' button is highlighted.

```
@@ -0,0 +1,22 @@
1 + #include <iostream>
2 + #include <list>
3 + #include <iterator>
4 + #include <cstdlib>
5 +
6 + int main(int argc, char* argv[]) {
7 +
8 +     std::list<int> intList;
9 +
10 +     for (int i = 1; i < argc; ++i) {
11 +         int temp = std::atoi(argv[i]);
12 +         intList.push_back(temp);
13 +     }
14 +
15 +     std::list<int>::iterator it = intList.begin();
16 +     while (it != intList.end()) {
17 +         std::cout << *it << std::endl;
18 +         it++;
19 +     }
20 +
21 +     return 0;
22 + }
```

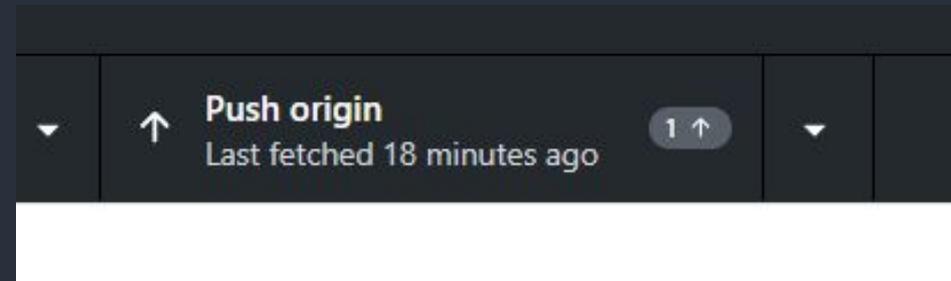
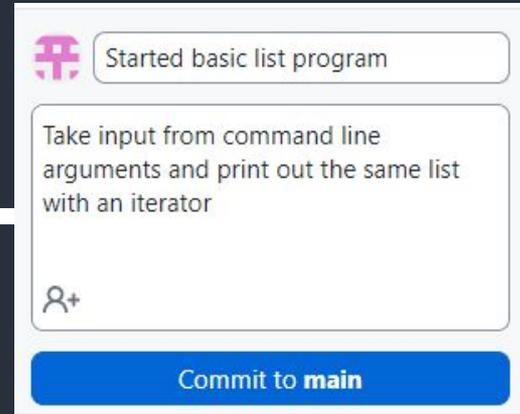
Started basic list program

Take input from command line arguments and print out the same list with an iterator

Commit to main

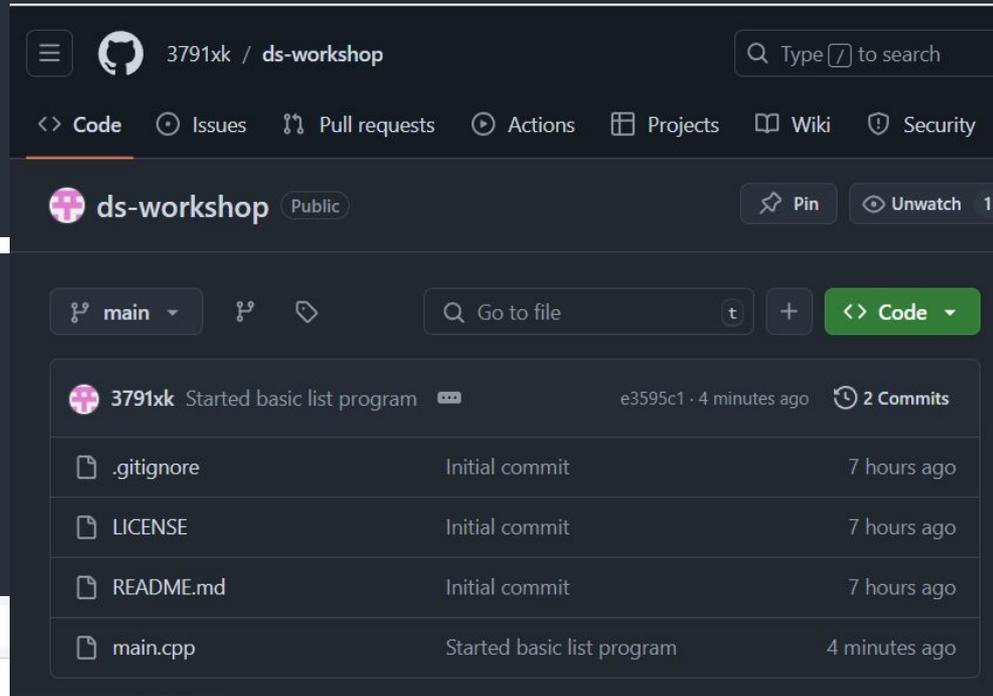
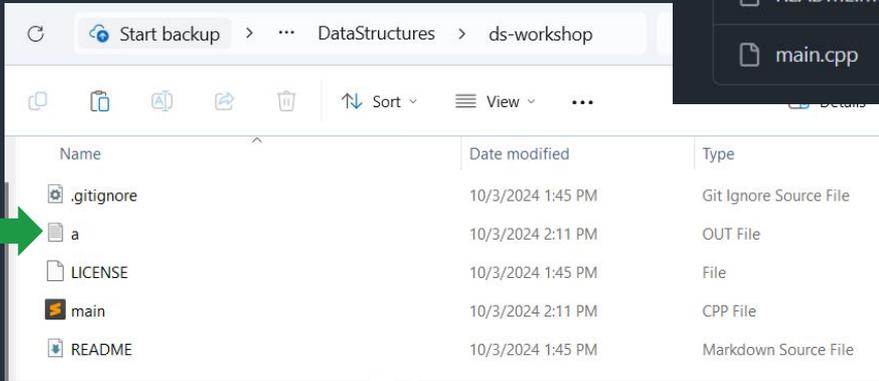
Commit to main

- Click the blue commit button
- Then click push to origin
- Your change will now be visible in your GitHub repository
- We'll find out why this isn't a best practice soon



Note the differences

- Your `.gitignore` file keeps compiled output files out of your repository



Branching

H

A

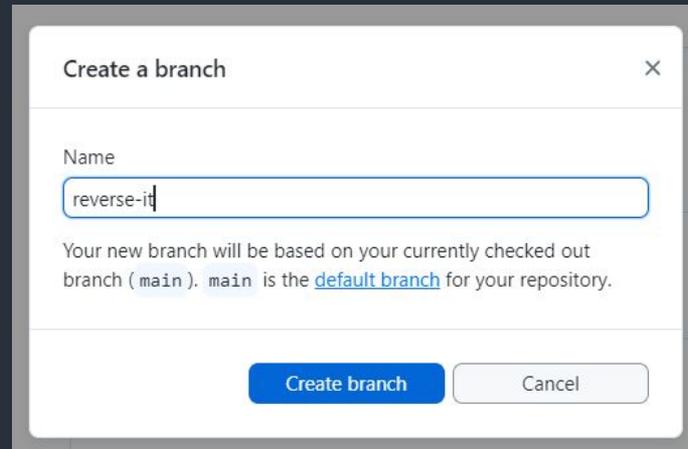
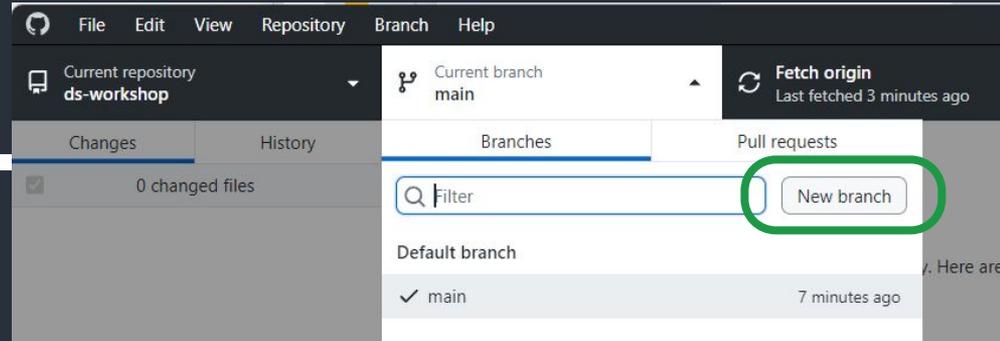
C

K



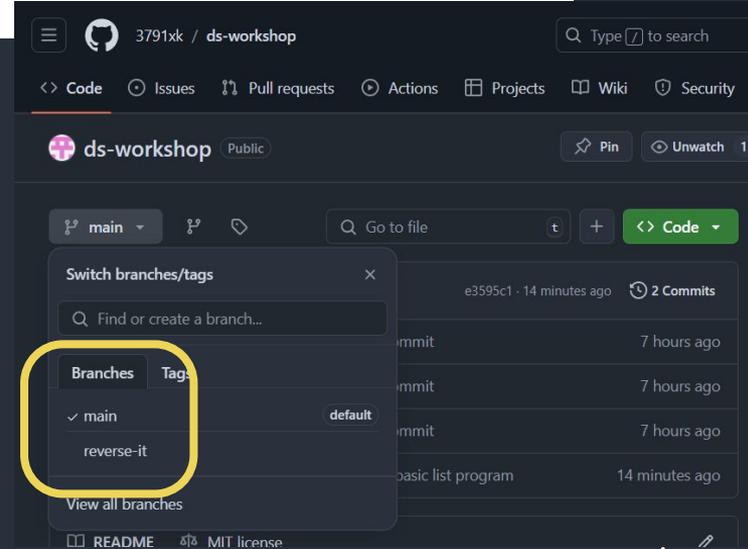
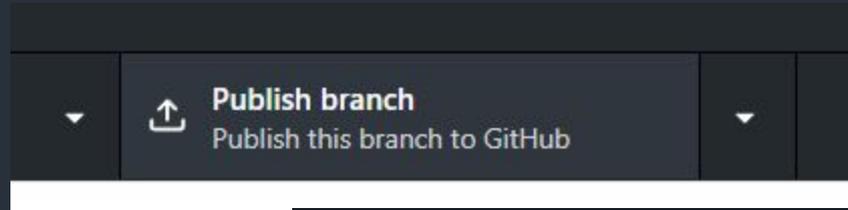
Make a branch

- A branch allows you to split off a version of the code to test things / add a feature
- ALWAYS pull repository first before creating
- Be specific when naming branch



See it on GitHub

- Click “Publish branch”
- You can commit to a branch as much as you want locally, but it’s good to push the branch to the repository to save it and for collaboration
- Then look at the repository page, right now everything is up to date with main



Reverse the iteration



- Swap the iterator out for a reverse iterator to iterate through the list backwards
- (Or just start at the end with the original iterator)

```
khusix@LAPTOP-NGACK2CL: ~$ g++ main.cpp -o ds.out -g -Wall -Wextra
khusix@LAPTOP-NGACK2CL: ~$ ./ds.out 1 2 3 5 8
8
5
3
2
1
khusix@LAPTOP-NGACK2CL: ~$
```

See the changes in GitHub Desktop

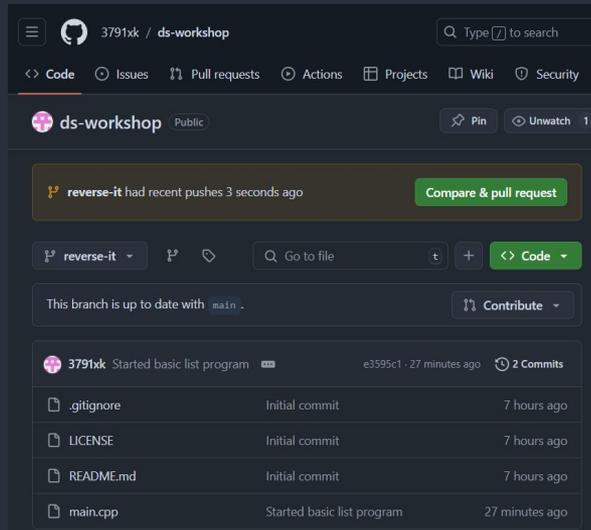
- Pretty straightforward, added parts are in green, removed parts are in red

```
@@ -12,8 +12,8 @@ int main(int argc, char* argv[]) {
    intList.push_back(temp);
}
- std::list<int>::iterator it = intList.begin();
- while (it != intList.end()) {
+ std::list<int>::reverse_iterator it = intList.rbegin();
+ while (it != intList.rend()) {
    std::cout << *it << std::endl;
    it++;
}
```

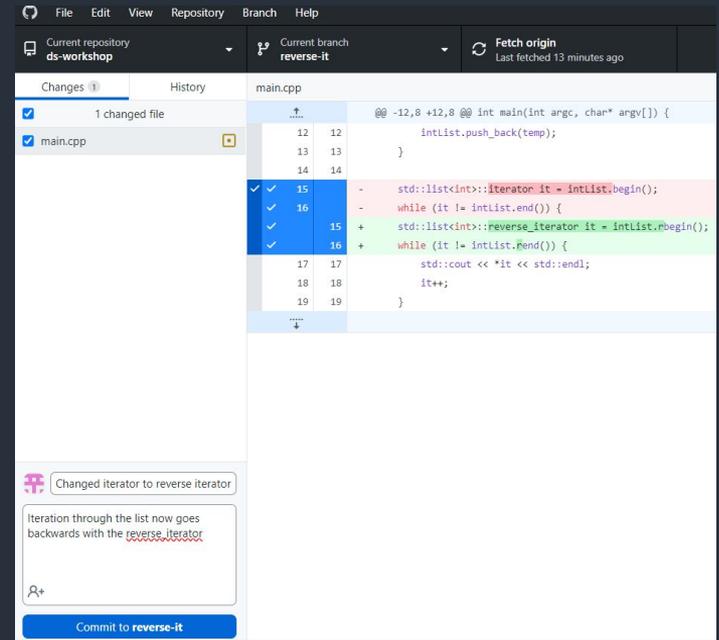


Commit your new changes

1. Write a commit name + description
2. Commit the changes to your branch
3. Push the changes to origin



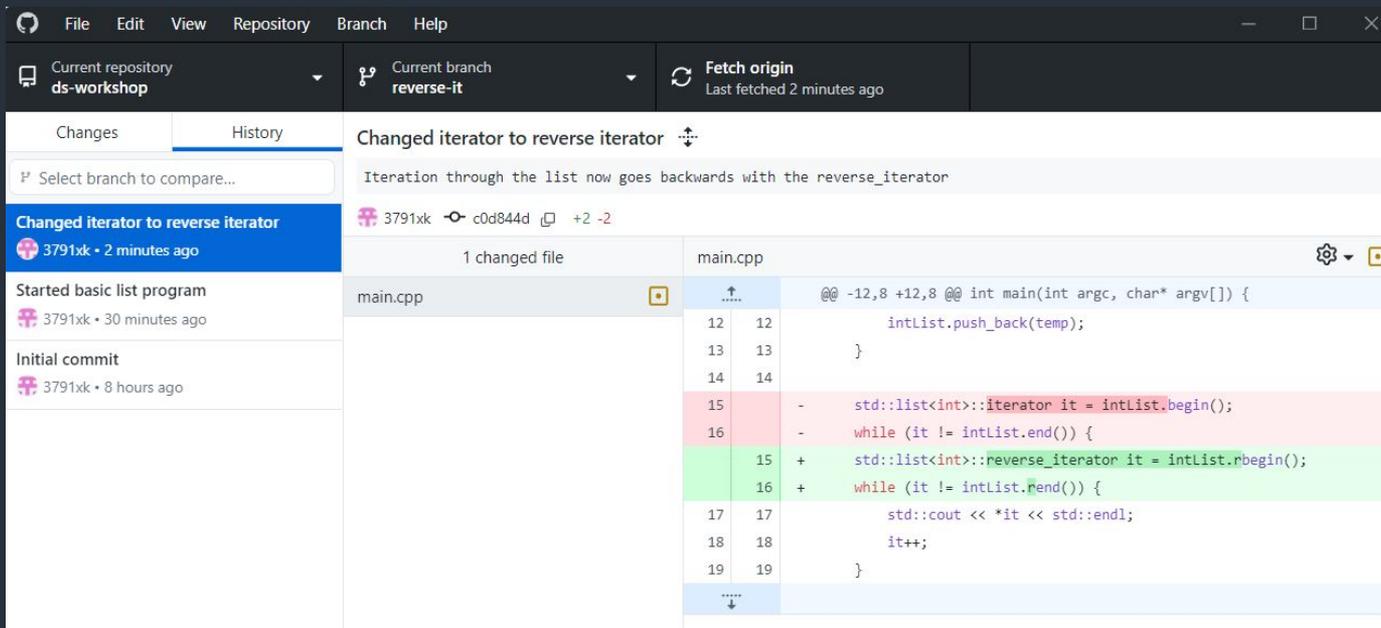
The screenshot shows the GitHub repository page for 'ds-workshop'. At the top, there's a navigation bar with 'Code', 'Issues', 'Pull requests', 'Actions', 'Projects', 'Wiki', and 'Security'. Below that, the repository name 'ds-workshop' is displayed with a 'Public' badge and 'Unwatch' button. A notification banner indicates that the 'reverse-it' branch has recent pushes. A 'Compare & pull request' button is visible. Below the notification, there's a section for the current branch 'reverse-it', showing it is up to date with the 'main' branch. A 'Contribute' button is also present. The commit history is shown, listing the initial commit for '.gitignore', 'LICENSE', and 'README.md', and the most recent commit for 'main.cpp' which started the basic list program.



The screenshot shows the VS Code interface with a diff view for the file 'main.cpp'. The 'Changes' pane on the left shows '1 changed file' and 'main.cpp'. The main editor area displays the code with a diff overlay. The diff shows that lines 15 and 16 have been changed. The original code (lines 15-16) used 'std::list<int>::iterator it = intList.begin();' and 'while (it != intList.end())'. The new code (lines 15-16) uses 'std::list<int>::reverse_iterator it = intList.rbegin();' and 'while (it != intList.rend())'. The diff highlights the changes in red and green. Below the code, there's a commit message input field with the text 'Changed iterator to reverse iterator' and a 'Commit to reverse-it' button.



See commit history



The screenshot shows the Visual Studio Code interface with the Git extension. The top bar displays the current repository as 'ds-workshop' and the current branch as 'reverse-it'. The 'Fetch origin' button indicates the last fetch was 2 minutes ago. The left sidebar shows the commit history with three entries:

- Changed iterator to reverse iterator** (3791xk • 2 minutes ago)
- Started basic list program** (3791xk • 30 minutes ago)
- Initial commit** (3791xk • 8 hours ago)

The main editor area shows a diff view for the selected commit. The title is 'Changed iterator to reverse iterator'. The description reads: 'Iteration through the list now goes backwards with the reverse_iterator'. Below this, it shows '1 changed file: main.cpp'. The diff view highlights the changes in the `main.cpp` file:

```
@@ -12,8 +12,8 @@ int main(int argc, char* argv[]) {
    intList.push_back(temp);
}
- std::list<int>::iterator it = intList.begin();
- while (it != intList.end()) {
+ std::list<int>::reverse_iterator it = intList.rbegin();
+ while (it != intList.rend()) {
    std::cout << *it << std::endl;
    it++;
}
```



Pull requests

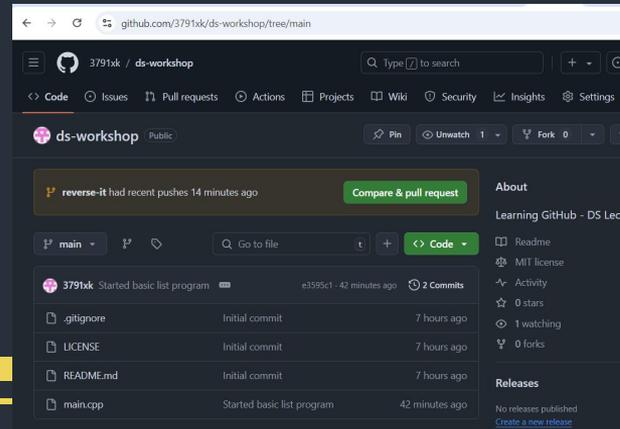
H **A** **C** **K**



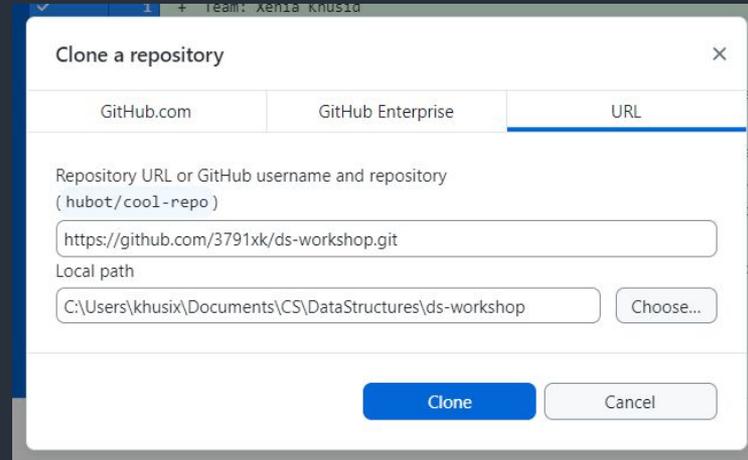
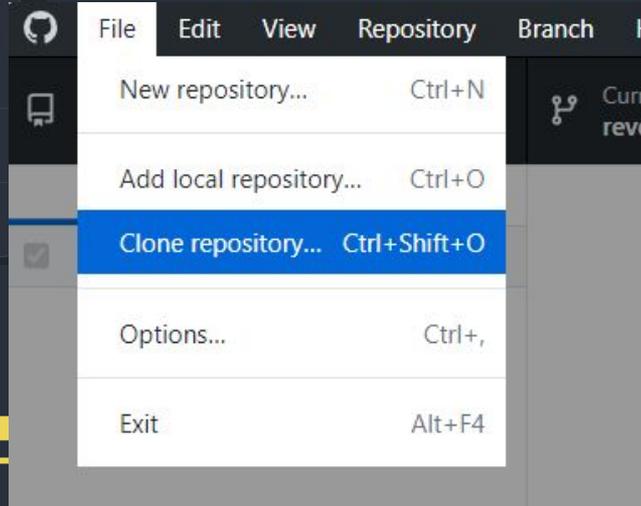
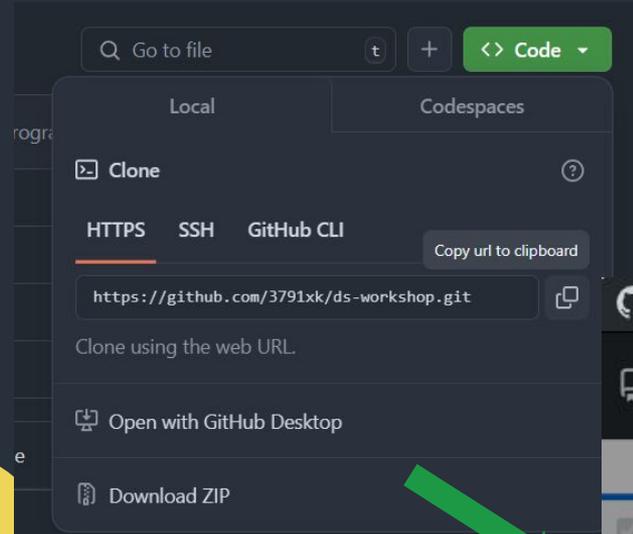
Collaborating With GitHub

- Introduce yourself with the person sitting next to you
 - Pairs or groups
 - Pass the repositories down
- We going to be collaborating on each other's websites
- Go to their GitHub repository

[https://github.com/3791xk/
ds-workshop](https://github.com/3791xk/ds-workshop)

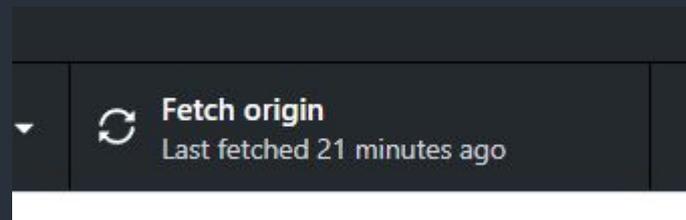
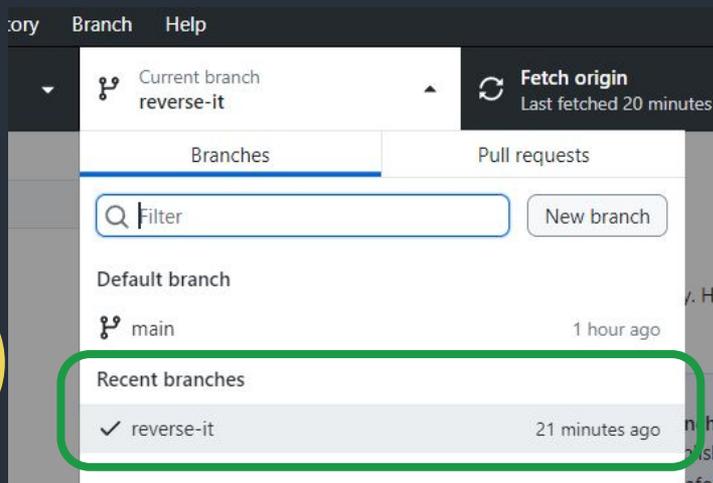


Clone their repository



Switch to the reverse branch

- Switch to their reverse branch
- Find the file in file explorer and open it in whatever IDE you use



Print out the square of the int



- Open up their code and make it print the squares of the integers instead of the integers

```
khusix@LAPTOP-NGACK2CL: ~$ g++ main.cpp -o ds.out -g -Wall -Wextra
khusix@LAPTOP-NGACK2CL: /mnt/c/Users/khusix/Documents/CS/DataStructures/ds-workshop$ ./ds.out 1 2 3 5 8
64
25
9
4
1
khusix@LAPTOP-NGACK2CL: /mnt/c/Users/khusix/Documents/CS/DataStructures/ds-workshop$
```

See the changes

```
main.cpp x
1  #include <iostream>
2  #include <list>
3  #include <iterator>
4  #include <cstdlib>
5
6  int main(int argc, char* argv[]) {
7
8      std::list<int> intList;
9
10     for (int i = 1; i < argc; ++i) {
11         int temp = std::atoi(argv[i]);
12         intList.push_back(temp);
13     }
14
15     std::list<int>::reverse_iterator it = intList.rbegin();
16     while (it != intList.rend()) {
17         std::cout << (*it)*(*it) << std::endl;
18         it++;
19     }
20
21     return 0;
22 }
```

Branch		Help	
Current branch	reverse-it	Fetch origin	Last fetched 27 minutes ago
main.cpp			
↑...		@@ -14,7 +14,7 @@ int main(int argc, char* argv[]) {	
14	14		
15	15		std::list<int>::reverse_iterator it = intList.rbegin();
16	16		while (it != intList.rend()) {
✓ ✓	17	-	std::cout << *it << std::endl;
✓ ✓	17	+	std::cout << (*it)*(*it) << std::endl;
18	18		it++;
19	19		}
20	20		
↓...			

Commit your changes

We hope you know how by now

1. Write a commit name + description
2. Commit the changes to your branch
3. Push the changes to origin

File Edit View Repository Branch Help

Current repository ds-workshop Current branch reverse-it Fetch origin Last fetched 29 minutes ago

Changes 1 History main.cpp

1 changed file

main.cpp

```
@@ -14,7 +14,7 @@ int main(int argc, char* argv[]) {
14 14
15 15     std::list<int>::reverse_iterator it = intList.rbegin();
16 16     while (it != intList.rend()) {
17 17 -     std::cout << *it << std::endl;
17 17 +     std::cout << (*it)*(*it) << std::endl;
18 18     it++;
19 19 }
20 20
```

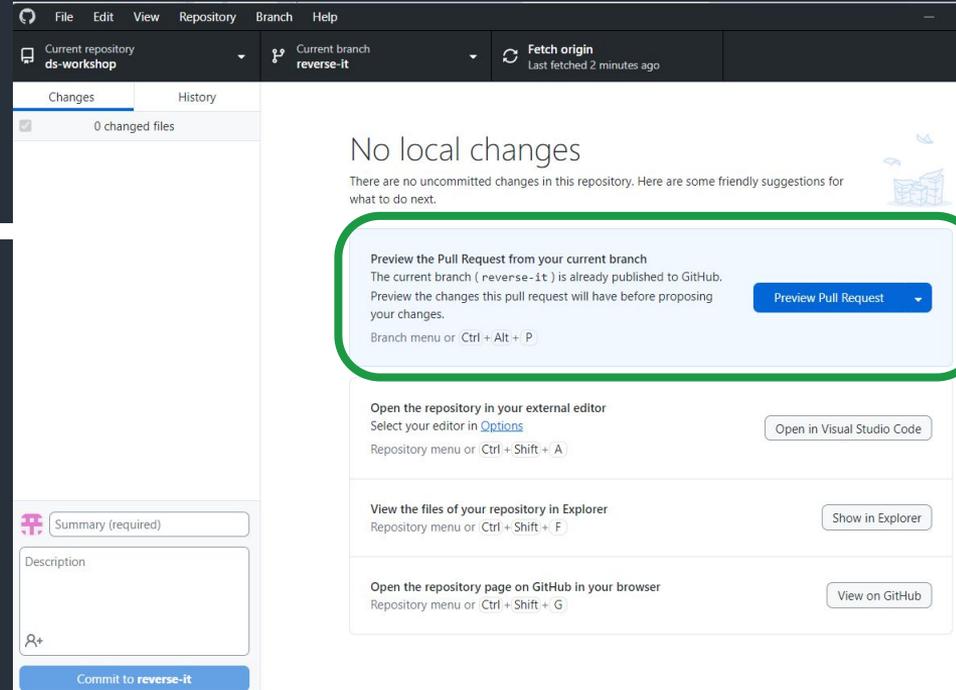
Print squares

Changes `cout` printing to square each number and print the square

Commit to reverse-it

Pull Requests

- Allows your team to view your branch on github for review and merging
- Pull Requests show up after you have Pushed Commits and will show up in blue
- Select Create Pull Request to create a pull request



Click “Create pull request”

Open a pull request

Merge 2 commits into `base: main` from `reverse-it`.

Showing changes from all commits

main.cpp

```
@@ -12,9 +12,9 @@ int main(int argc, char* argv[]) {
12 12     intList.push_back(temp);
13 13     }
14 14
15 -   std::list<int>::iterator it = intList.begin();
16 -   while (it != intList.end()) {
17 -       std::cout << *it << std::endl;
15 +   std::list<int>::reverse_iterator it = intList.rbegin();
16 +   while (it != intList.rend()) {
17 +       std::cout << (*it)*(*it) << std::endl;
18 18     it++;
19 19     }
20 20

```

✓ Able to merge. These branches can be automatically merged.

Create pull request Cancel

Creating a pull request

- Write a title and description
- See the commits
- See changes from the commits
- This is a request to have your changed merged

The screenshot shows the GitHub interface for a repository. At the top, it says "3791xk / ds-workshop". Below that, there are navigation links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The main content area is titled "Open a pull request" and includes a description: "Create a new pull request by comparing changes across two branches. If you need to, you can also compare across forks. Learn more about...". Below this, there are fields for "base: main" and "compare: reverse-it", with a green checkmark indicating "Able to merge".

The "Add a title" section has a text input field containing "Reverse and square the integers". Below that, the "Add a description" section has a rich text editor with the text: "The list is iterated over in reversed order using a reverse iterator and each value is squared before it's printed." There are also buttons for "Markdown is supported" and "Paste, drop, or click to add files".

At the bottom right, there is a green button labeled "Create pull request".

The commit history shows two commits on Oct 3, 2024:

- Changed iterator to reverse iterator (3791xk committed 46 minutes ago)
- Print squares (3791xk committed 15 minutes ago)

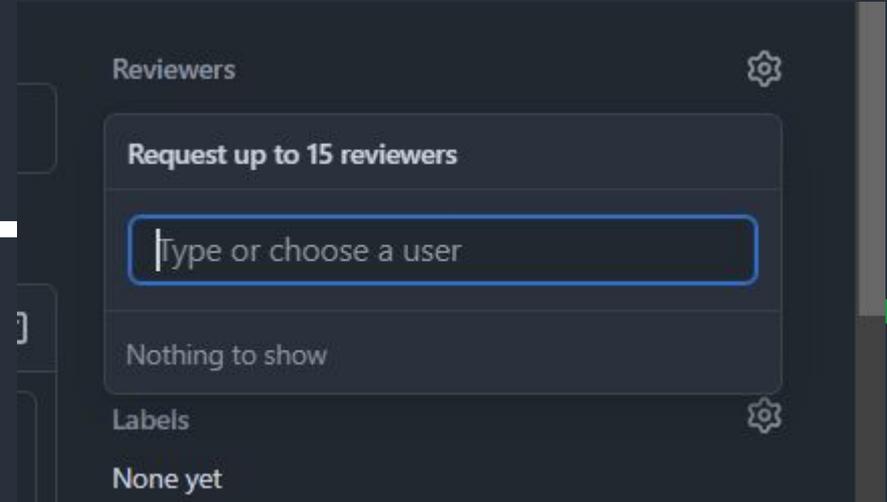
The diff view shows changes to main.cpp:

```
@@ -12,9 +12,9 @@ int main(int argc, char* arg
12      intList.push_back(temp);
13  }
14
15  - std::list<int>::iterator it = intList.begin();
16  - while (it != intList.end()) {
17  -     std::cout << *it << std::endl;
15  + std::list<int>::reverse_iterator it = intList.rbegin();
16  + while (it != intList.rend()) {
17  +     std::cout << (*it)*(*it) << std::endl;
18
19  +     it++;
19  }
20
```



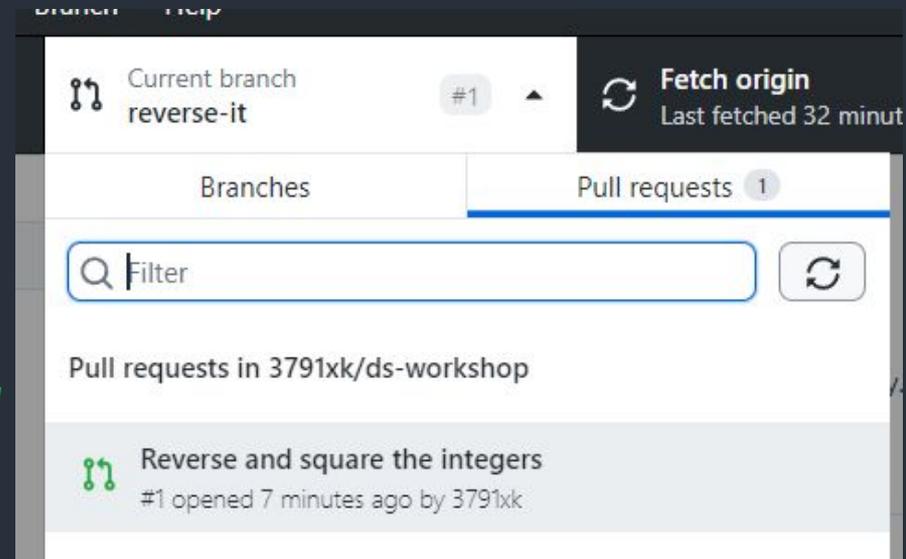
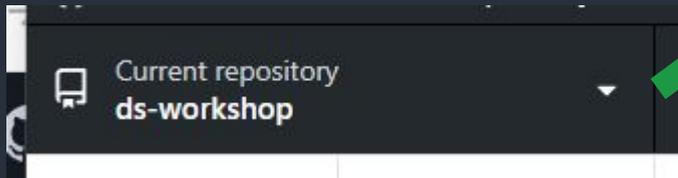
Request a review

- Request the owners username to review
- Your partner can make comments on, request changes, close, or approve and merge a pull request.
- Click on the 'Files Changed' button to review the PR



Switch to your repository on GitHub Desktop

- Switch “Current repository” to your own repo
- Switch to the “Pull request” branch and fetch origin



Review the Pull Request

- Run their code to make sure it actually works
- Review on GitHub
- Optionally add comments
- Merge when ready, it'll turn purple

The screenshot shows a GitHub Pull Request (PR) titled "Reverse and square the integers #1". The PR is currently in a "Review" state, indicated by a green "Open" button. The PR description states: "The list is iterated over in reversed order using a reverse iterator and each value is squared before it's printed." The PR was created by user "3791xk" and is targeting the "main" branch from the "reverse-it" branch. The PR includes 2 commits: "changed iterator to reverse iterator" and "print squares". The PR is currently open for review, as indicated by the "Open" button and the "Merge pull request" button being highlighted with a yellow box. The PR is currently in a "Review" state, indicated by the "Open" button and the "Merge pull request" button being highlighted with a yellow box. The PR is currently in a "Review" state, indicated by the "Open" button and the "Merge pull request" button being highlighted with a yellow box. The PR is currently in a "Review" state, indicated by the "Open" button and the "Merge pull request" button being highlighted with a yellow box.

Reverse and square the integers #1

Open 3791xk wants to merge 2 commits into main from reverse-it

Conversation 0 Commits 2 Checks 0 Files changed 1 +3 -3

3791xk commented 2 minutes ago (Owner)

The list is iterated over in reversed order using a reverse iterator and each value is squared before it's printed.

3791xk added 2 commits 1 hour ago

- changed iterator to reverse iterator (c8d844d)
- print squares (2c5df6e)

Require approval from specific reviewers before merging (Rulesets ensure specific people approve pull requests before they're merged. Add rule)

Continuous integration has not been set up (GitHub Actions and several other apps can be used to automatically catch bugs and enforce style.)

This branch has no conflicts with the base branch (Merging can be performed automatically.)

Merge pull request (You can also open this in GitHub Desktop or view command line instructions.)

Add a comment

Reviews: No reviews. Still in progress? Convert to draft.

Assignees: No one—assign yourself.

Labels: None yet.

Projects: None yet.

Milestone: No milestone.

Development: Successfully merging this pull request may close these issues. None yet.

Notifications: Unsubscribe. You're receiving notifications because you're watching this repository.

Questions/
Comments?



HackRPI 2024

November 9-10th

<https://hackrpi.com/>

How to HackRPI

October 4th

 Instagram: @hack.rpi

 LinkedIn: @HackRPI

 Website: hackrpi.com