CS221 - Lab01

Paul Haskell

INTRODUCTION:

In this class the instructors and students will use the **git** tools and **GitHub** cloud repository to exchange documents and completed programs. Everyone in this class should have used these before—if you have not, please see the instructor to get some helpful background material.

Recall you can commit multiple versions of a file or document with git, and you can push multiple versions of a file or document to GitHub. Git and GitHub will store all the versions, and you and the instructors can retrieve any particular version. For assignments, unless you ask otherwise, the instructor will look at the newest version you pushed **before** the deadline time.

For the CS221 class, each of you will work with two repositories:

- A shared one called "CourseInfo" that the instructor and TA will populate with documents, sample software, assignments, etc. You each will make a local copy on your own computer, and you will update your local copy when the instructor tells you there are updates on the CourseInfo repository's cloud server. All students can read this repository, and the TA and instructor can read and write it.
- An individual repository for each of you that you will use to turn in your work. You will write, debug, and test software on your personal computer and then will "push" it to your individual repository on the Cloud server. The TA and instructor then will fetch your software, test it, review it, and grade it. Other students cannot read or write your repository.

In CS221 you also will work with a terminal program called "Git BASH". If you don't have this program installed on your computer already, simply install a new version of the git tools from install Git BASH , being sure to download and install the 'Git BASH' tool.

- on Windows, during installation please select the "Add a Git Bash Profile to Windows Terminal" checkbox. This lets you open Git BASH from the terminal window, instead of 'cmd' or PowerShell.

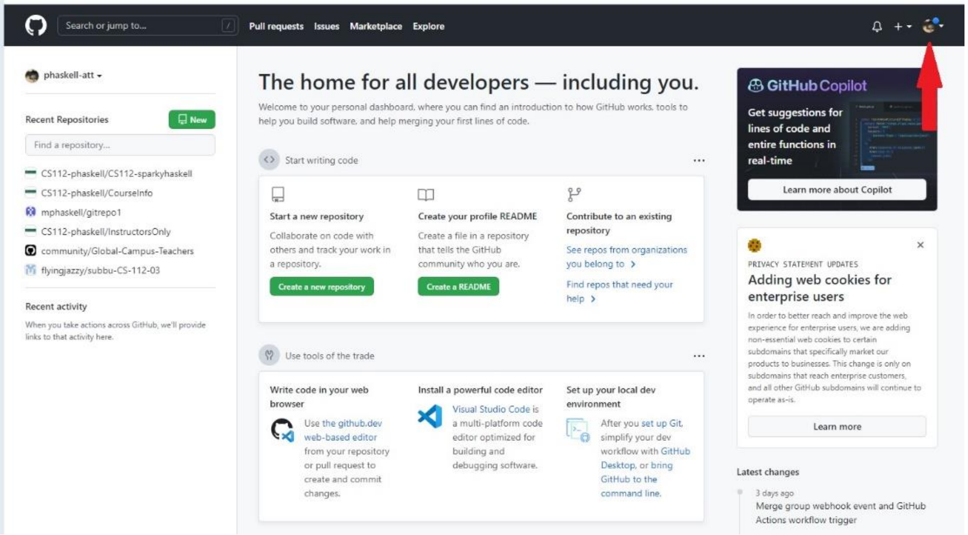
LAB01:

Most of you should have a GitHub account already. If you do, please email it to the instructor, or simply walk up and tell him what it is. He will add you to the "CS221 organization". Please accept any email invites you get from GitHub related to CS221.

If you do not have a GitHub account yet, please see the instructor so you can set one up together.

GitHub stores a Personal Access Token on your computer so you can exchange information with GitHub without having to type a password every time. However, your old Personal Access Token may have expired. If so, please follow these steps to create a new one:

    - In your web browser, log into GitHub.
    - Click on the little round button in the upper right, and select the "Settings" option


Search or jump to...
Pull requests
Issues
Marketplace
Explore
phaskell-att
Recent Repositories
New
Find a repository...
CS112-phaskell/CS112-sparkyhaskell
CS112-phaskell/CourseInfo
mphaskell/gitrepo1
CS112-phaskell/InstructorsOnly
community/Global-Campus-Teachers
flyingjazzy/subbu-CS-112-03
Recent activity
Start writing code
Start a new repository
Create a new repository
Create your profile README
Create a README
Contribute to an existing repository
See repos from organizations you belong to
Find repos that need your help
Use tools of the trade
Write code in your web browser
Install a powerful code editor
Set up your local dev environment
GitHub Copilot
Learn more about Copilot
PRIVACY STATEMENT UPDATES
Learn more
Latest changes

The home for all developers — including you.

Welcome to your personal dashboard, where you can find an introduction to how GitHub works, tools to help you build software, and help merging your first lines of code.

When you take actions across GitHub, we'll provide links to that activity here.

Collaborate on code with others and track your work in a repository.

Create a file in a repository that tells the GitHub community who you are.

Use the github.dev web-based editor from your repository or pull request to create and commit changes.

Visual Studio Code is a multi-platform code editor optimized for building and debugging software.

After you set up Git, simplify your dev workflow with GitHub Desktop, or bring GitHub to the command line.

Get suggestions for lines of code and entire functions in real-time

Adding web cookies for enterprise users

In order to better reach and improve the web experience for enterprise users, we are adding non-essential web cookies to certain subdomains that specifically market our products to businesses. This change is only on subdomains that reach enterprise customers, and all other GitHub subdomains will continue to operate as-is.

3 days ago
Merge group webhook event and GitHub Actions workflow trigger

    - On the bottom of the left side, select "Developer Settings"
    - On the left, select "Personal access tokens" ("classic" not "beta")
    - Set the "Expiration" to be some date after the class ends
    - Select at least the following "scopes": repo, admin:org, user, delete_repo
    - Then select "Generate Token". Copy the resulting token and paste it into a text document, so you have a copy.
- The first time git prompts you for your **password**, paste in the **Personal Access Token**. This should let the command execute successfully. And the token should be stored in your "Credential Manager", so from now on, you can run git commands without being asked for a password.

1) DOWNLOAD THE CourseInfo REPOSITORY TO YOUR COMPUTER:

Do this after you have accepted the invite to the CourseInfo repository.

- Run a BASH terminal.
- Create a directory on your personal PC to store the CS221 course info and also your coursework. You will use this directory for the entire course—please create only one!
    - In a convenient place on your computer's filesystem, like your home directory, make a directory called "CS221" by typing: `mkdir CS221`
    - Now have the terminal window enter the directory: type: `cd CS221`
      Make sure your terminal is now inside the `CS221` directory!

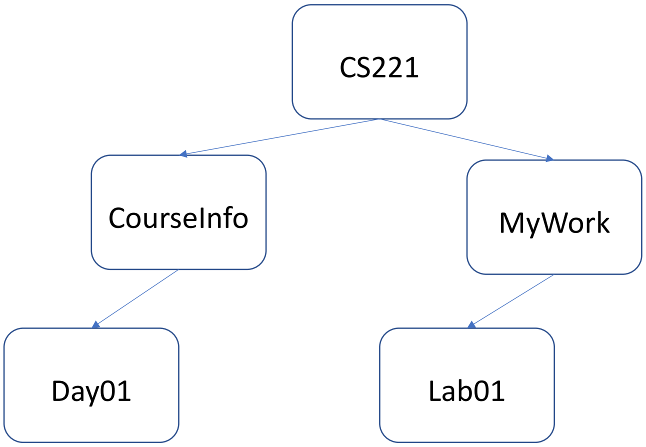
- Clone material in the CourseInfo repository to your local computer: from the **CS221** directory, in your terminal or command window, type:

```
git clone https://github.com/CS221-phaskell/CourseInfo CourseInfo
```

- If you are asked to authenticate with your browser or with your Personal Access Token, do so.

2) SET UP A DIRECTORY ON YOUR PERSONAL COMPUTER TO STORE YOUR COURSEWORK

- In your **CS221** directory (not inside the **CourseInfo** subdirectory), make a directory called **MyWork**
- Copy the **.gitignore** file from the **CourseInfo/Day01** directory to your **MyWork** directory.
  - In your BASH terminal, cd to the **CourseInfo/Day01** directory and enter
    ```
    cp .gitignore ../../MyWork/.gitignore
    ```
  - The **ls** command hides any filenames that start with a '.' . To see those files, enter:
    ```
    ls -a
    ```
- Inside **MyWork**, create a subdirectory called **Lab01**. (In later labs you will create directories called Lab02, Lab03, …)

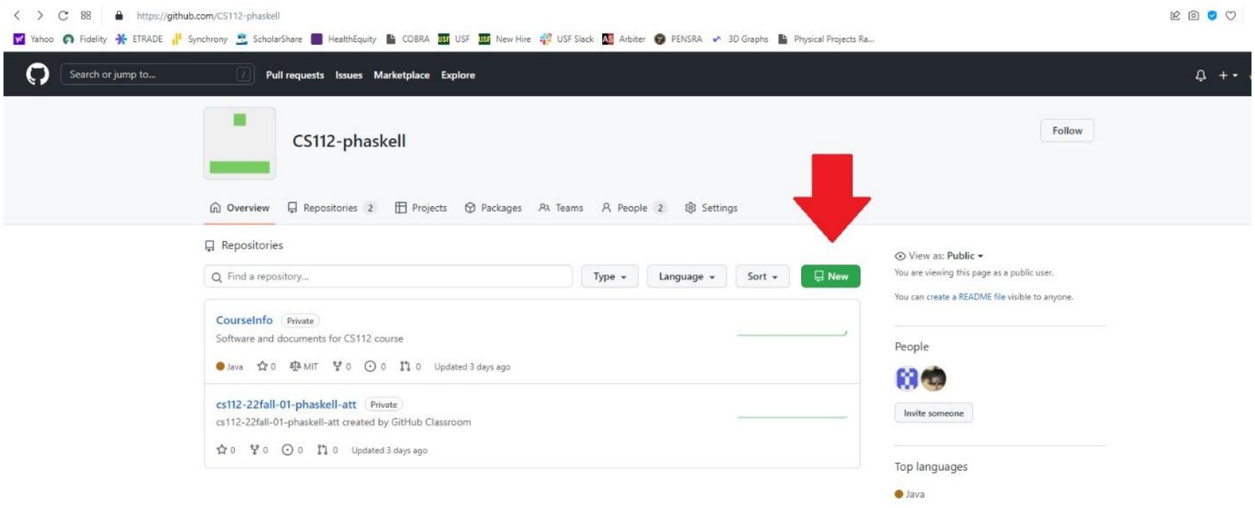


- Inside your **Lab01** directory, create a Bash script called **myinfo.sh** . When this script is run, it should print (using the `echo` command):
  - Your name
  - Your USF email address
  - Your GitHub username
  - Then, enter some interesting but not overly personal information about yourself, to help the TA and Paul learn all your names more quickly
  - Please be sure to save this file as a ".sh" file, not a PDF, txt, etc. Get help if you need it.
- Also inside **Lab01**, please add a JPEG photo of your face (to help the TA and Paul learn to match your faces with your names)
- You can use a graphical text editor to create this script, or you can use one of the command-line text editors.
- Be sure to run your script from your Bash terminal, to be sure it works

SET UP YOUR PERSONAL GITHUB REPOSITORY AND TIE IT TO YOUR “MyWork” DIRECTORY

Now you will create your personal GitHub repository and tie your **MyWork** directory to it. Then you will push your **myinfo.sh** and JPEG file to GitHub.

- (If you closed your github.com webpage already, log into the github.com website.)
- (If your github.com screen is not saying “Create your first CS221-phaskell repository, then click on the little round button in the upper right, select the “Your Organizations” option, and select “CS221-phaskell”.)
- Click the “New” or “Create a new repository” button to create a new repository.



https://github.com/CS112-phaskell

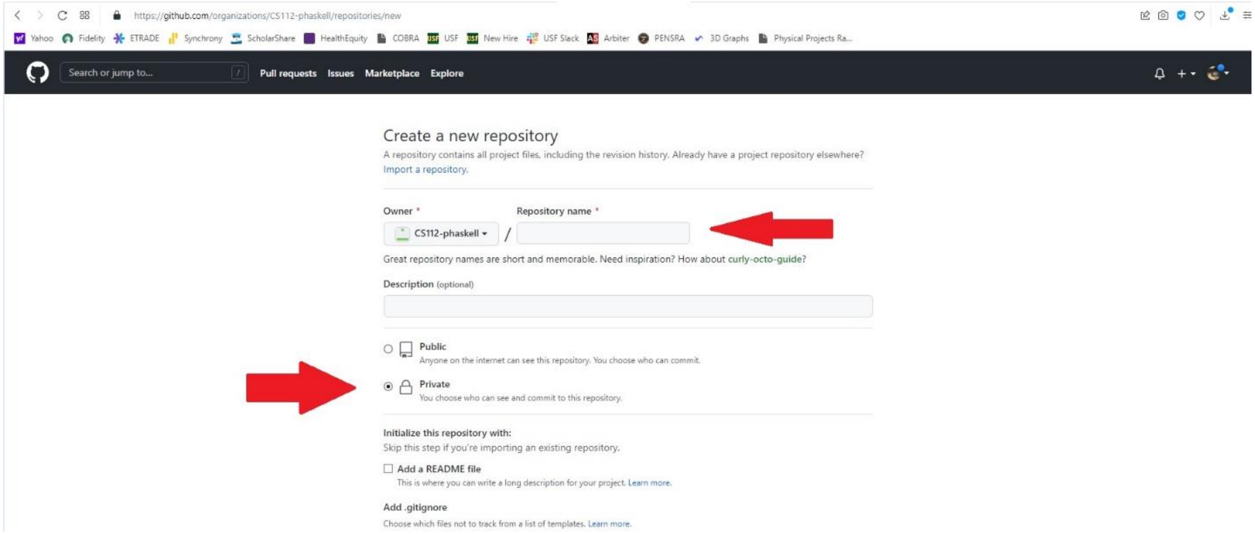
CourseInfo Private
Software and documents for CS112 course
Java ☆ 0 MIT 0 0 0 Updated 3 days ago

cs112-22fall-01-phaskell-att Private
cs112-22fall-01-phaskell-att created by GitHub Classroom
☆ 0 0 0 0 Updated 3 days ago

You are viewing this page as a public user.

You can create a README file visible to anyone.

- Name your repository “CS221-<<GitHub UserID>>”
    - For example, CS221-stephcurry
    - It is very important you get the format of this name correct, including capitalization!
- Make it PRIVATE – not public
- Make sure that the Owner is “CS221-phaskell”



https://github.com/organizations/CS112-phaskell/repositories/new

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner * Repository name *

CS112-phaskell /

Great repository names are short and memorable. Need inspiration? How about curly-octo-guide?

Description (optional)

○ 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:
Skip this step if you're importing an existing repository.

□ Add a README file
This is where you can write a long description for your project. Learn more.

Add .gitignore
Choose which files not to track from a list of templates. Learn more.

- Click the “Create Repository” button. Don’t close the browser window! And don’t run any of the offered “quick setup” steps.
- In a terminal window, change to the **MyWork** directory (If you are in the **Lab01** directory, type `cd ..`) Make sure your **myinfo.sh** file and your JPEG file are inside the **Lab01** directory.
    - Type: `git init -b main`
    - Type: `git add Lab01`
    - Now “commit” your added directories and files: type (include the double-quote marks)
        - `git commit -m "My first commit for Lab01"`
        - The words after the ‘-m’ are a comment for this git commit
    - On the webpage with your GitHub repository, grab the URL for your repository from the webpage titlebar: probably something like **https://github.com/CS221-phaskell/CS221-YourGitHubID**
    - In the terminal window type:
      `git remote add origin <<TheURLYouCopied>>`
    - Now send your file to your repository storage in the cloud: type:
      `git push --set-upstream origin main`
    - That’s it! You now have a personal repository set up for the class, and you have uploaded your **Lab01** directory. Use your browser to look at your repository, and you should see your files.

CONCLUSION:

The deadline for completion of Lab01 is <u>11:59pm Friday January 24</u>.

The main goals of this lab are for you to install software that will be required for the class, to set up your GitHub repositories, and to follow the lab instructions accurately to demonstrate you succeeded in the first two goals. Future labs will have more involved scoring rubrics, and a portion of their scoring will be based on instructor-determined standards of Design Quality and Code Quality. For this first lab, the scoring rubric is simpler:

Task	Score, points
Created a personal GitHub repository with the correct name	1
Created properly named **Lab01** subdirectory	1
Created properly named **myinfo.sh** file	1
Created properly named JPEG file	1
myinfo.sh runs and prints all the requested information	4