# Github Classroom and IntelliJ

Please follow the directions below EXACTLY to complete this recitation successfully.

1. Head over to https://github.com and sign up for an account. Please USE YOUR DESU EMAIL ACCOUNT when signing up for the account.

2. If you do not have a recent version of Java, please install an OpenJDK version, which you can download from JDK Builds from Oracle (java.net). Please select and install the "Ready for use" JDK. A good one to use is jdk.java.net

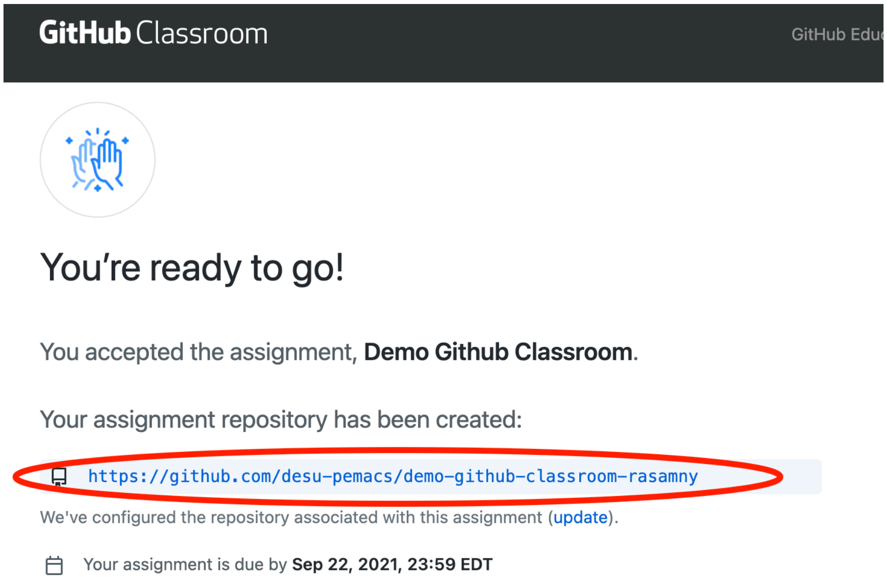
3. If you have not install IntelliJ, go to IntelliJ Download and download and install the IntelliJ IDE Community Edition. This is the free version of the IDE.

4. You will next need to accept the assignment in Github classroom and link it to your Github account. Go to your Blackboard course page and click on the assignment link under the Skill Builder 0 on the Skill Builder page

5. Find your name/email in the list and select it. IT IS IMPORTANT THAT YOU SELECT YOUR NAME FROM THE LIST!!!!!

6. Accept the assignment and wait a minute before refreshing the page. You should now have a link to your own private repository that contains the assignment.

7. Click the link shown on the page. Should you need the link in the future, simply click on the exercise link provided above to come back to this page.



You're ready to go!

You accepted the assignment, **Demo Github Classroom**.

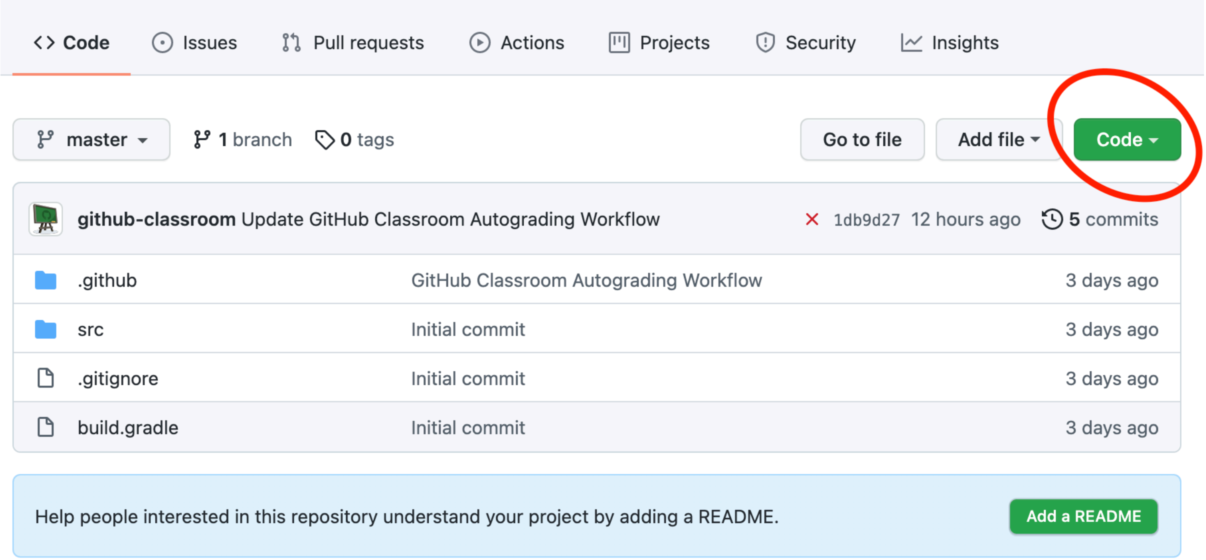
Your assignment repository has been created:

https://github.com/desu-pemacs/demo-github-classroom-rasamny

We've configured the repository associated with this assignment (update).

Your assignment is due by **Sep 22, 2021, 23:59 EDT**

8. You should now see your private repository with a green Code button on the top right as shown in the image below.

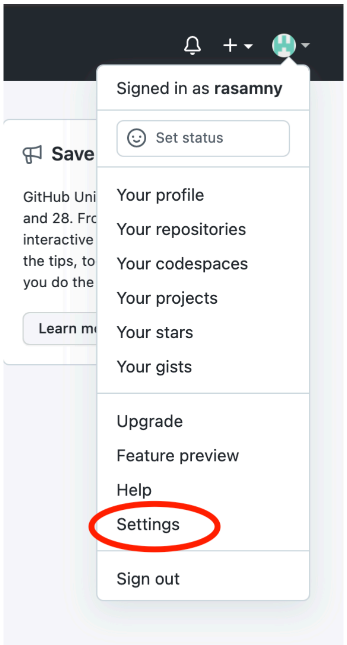


**github-classroom** Update GitHub Classroom Autograding Workflow ✕ 1db9d27 12 hours ago **5** commits

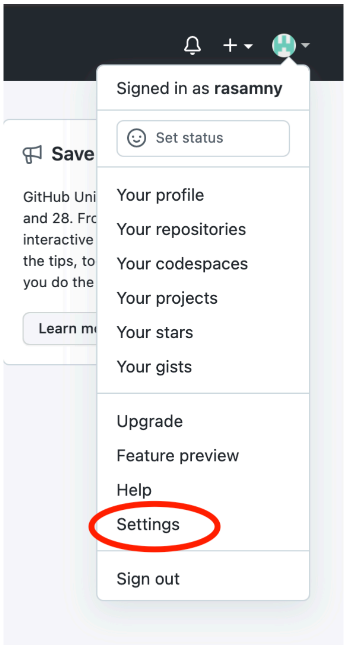
| | | |
|---|---|---|
| .github | GitHub Classroom Autograding Workflow | 3 days ago |
| src | Initial commit | 3 days ago |
| .gitignore | Initial commit | 3 days ago |
| build.gradle | Initial commit | 3 days ago |

Help people interested in this repository understand your project by adding a README.

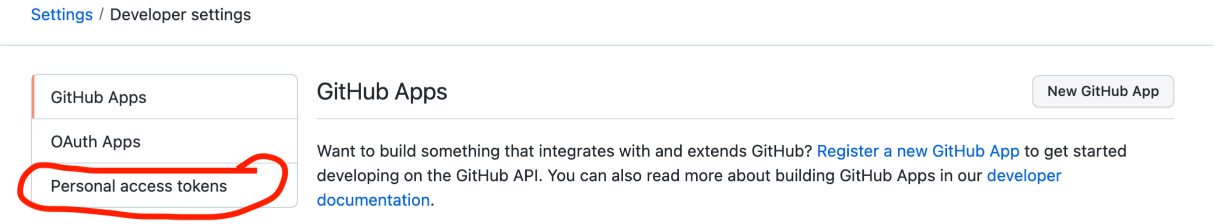
9. You will now need to create a Github authentication token to be used in IntelliJ. Click on your avatar on the top right of the screen. You should see a drop-down menu as shown in the image below. Select Settings.



10. On the Settings page, click on the Developer Setting menu item on the left as shown below.

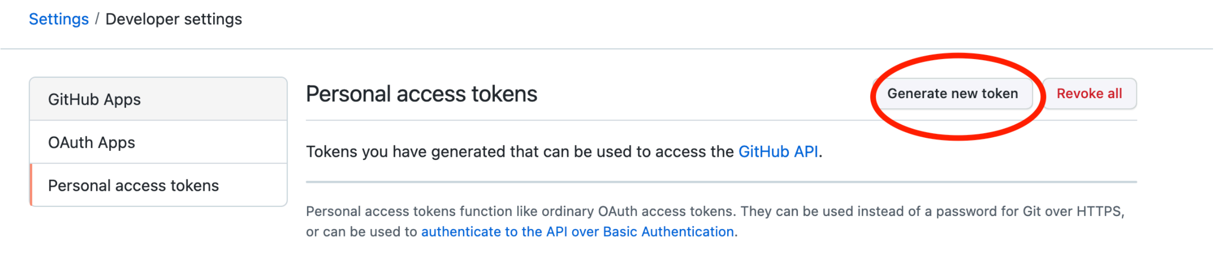


11. On the next page, select the Personal Access Tokens as shown in the image below.



Want to build something that integrates with and extends GitHub? Register a new GitHub App to get started developing on the GitHub API. You can also read more about building GitHub Apps in our developer documentation.

12. Then click on the Generate new token button on the top right as shown in the image below.



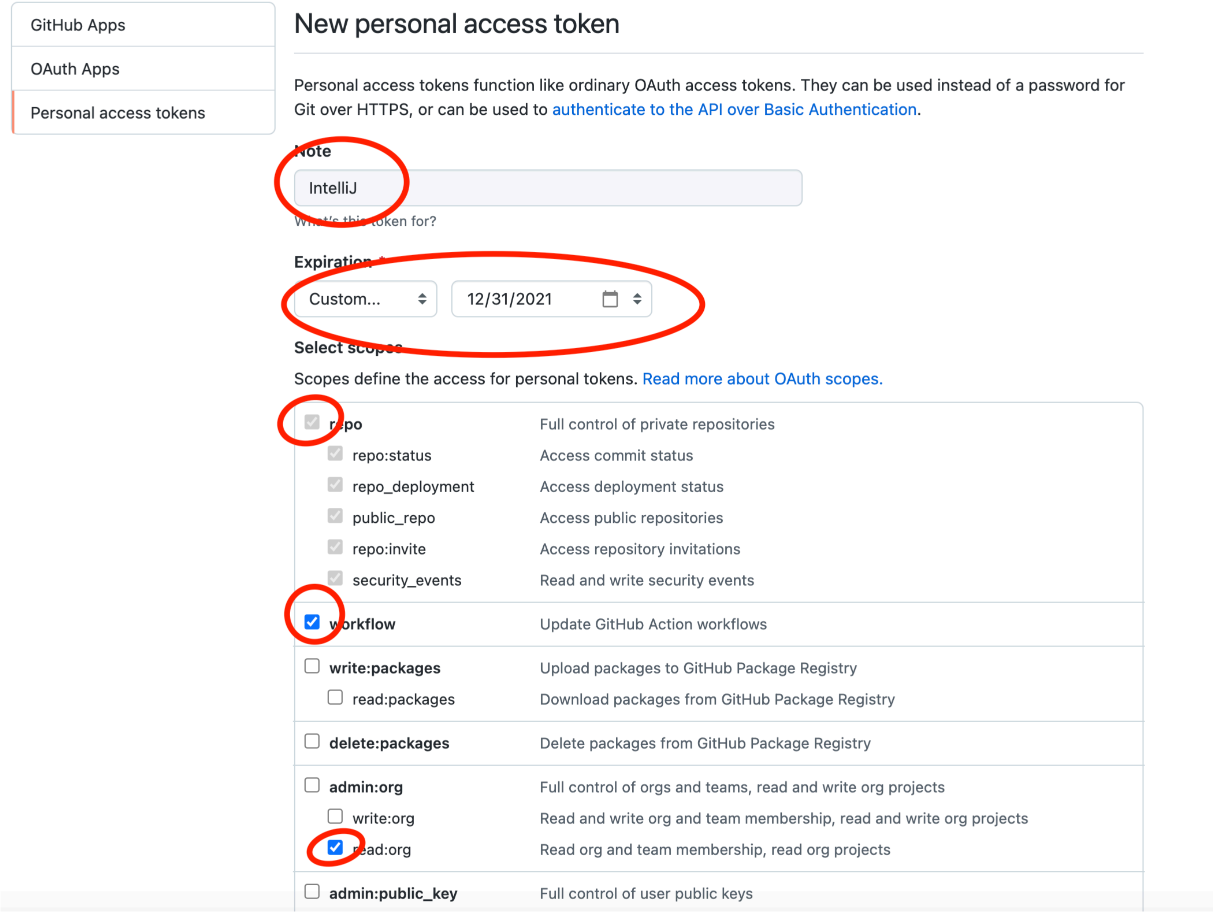
Settings / Developer settings

Personal access tokens

Tokens you have generated that can be used to access the GitHub API.

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.

13. Enter the word IntelliJ in the Note textbox and select the repo, workflow, gist, and read:org checkboxes as shown in the image below. Also, for the expiration, select custom and enter December 31, 2021. This will make your token valid for the entire course.



New personal access token

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.

Note

IntelliJ

What's this token for?

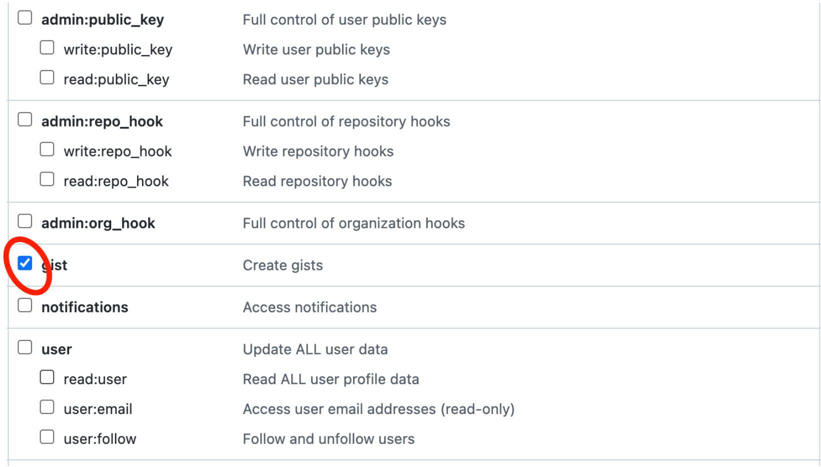
Expiration *

Custom... 12/31/2021

Select scopes

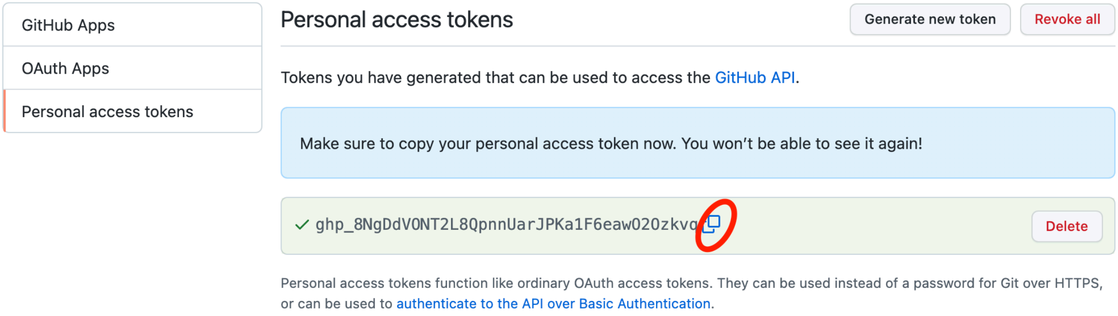
Scopes define the access for personal tokens. Read more about OAuth scopes.

| | |
|---|---|
| ☑ **repo** | Full control of private repositories |
| ☑ repo:status | Access commit status |
| ☑ repo_deployment | Access deployment status |
| ☑ public_repo | Access public repositories |
| ☑ repo:invite | Access repository invitations |
| ☑ security_events | Read and write security events |
| ☑ **workflow** | Update GitHub Action workflows |
| ☐ **write:packages** | Upload packages to GitHub Package Registry |
| ☐ read:packages | Download packages from GitHub Package Registry |
| ☐ **delete:packages** | Delete packages from GitHub Package Registry |
| ☐ **admin:org** | Full control of orgs and teams, read and write org projects |
| ☐ write:org | Read and write org and team membership, read and write org projects |
| ☑ read:org | Read org and team membership, read org projects |
| ☐ **admin:public_key** | Full control of user public keys |



| | |
|---|---|
| □ admin:public_key | Full control of user public keys |
| □ write:public_key | Write user public keys |
| □ read:public_key | Read user public keys |
| □ admin:repo_hook | Full control of repository hooks |
| □ write:repo_hook | Write repository hooks |
| □ read:repo_hook | Read repository hooks |
| □ admin:org_hook | Full control of organization hooks |
| ☑ gist | Create gists |
| □ notifications | Access notifications |
| □ user | Update ALL user data |
| □ read:user | Read ALL user profile data |
| □ user:email | Access user email addresses (read-only) |
| □ user:follow | Follow and unfollow users |

14. Click the green Generate Token button at the bottom of the screen.
15. Copy the token by clicking on the copy button as indicated in the image below. You should store the token somewhere because you will not be able to recover it. If you should forget it, you will need to regenerate a new token.



Personal access tokens

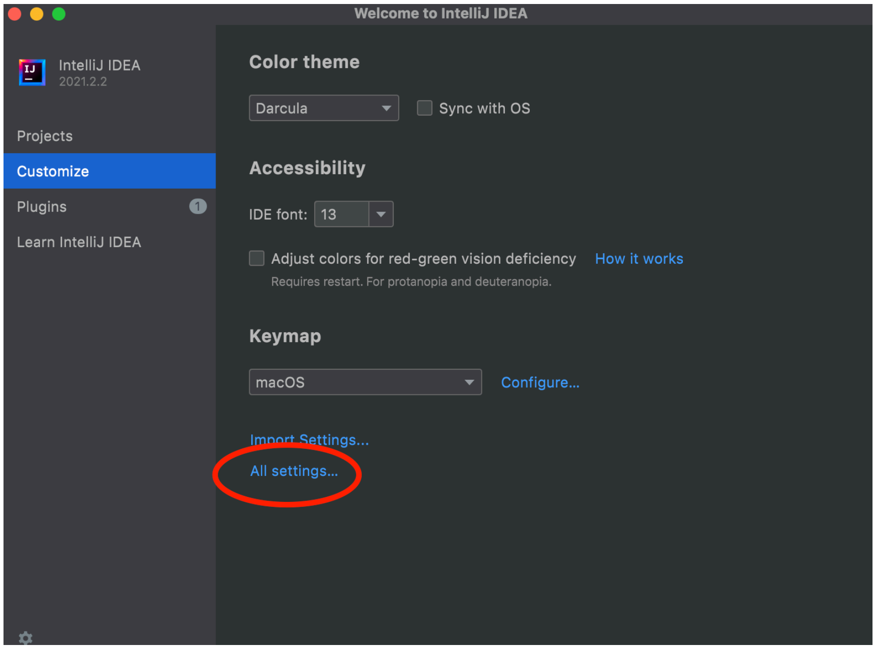
Tokens you have generated that can be used to access the GitHub API.

Make sure to copy your personal access token now. You won't be able to see it again!

✓ ghp_8NgDdV0NT2L8QpnnUarJPKa1F6eaw020zkvq

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.

16. Now, open up IntelliJ and click on Customize and then All Settings as shown in the image below.



Color theme

Darcula □ Sync with OS

Accessibility

IDE font: 13

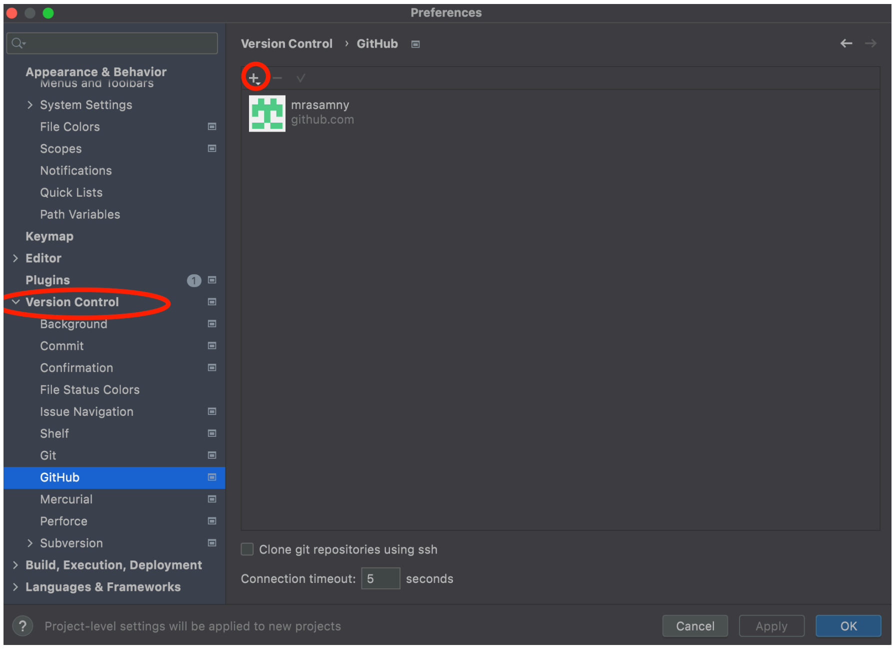
□ Adjust colors for red-green vision deficiency How it works

Requires restart. For protanopia and deuteranopia.

Keymap

macOS Configure...

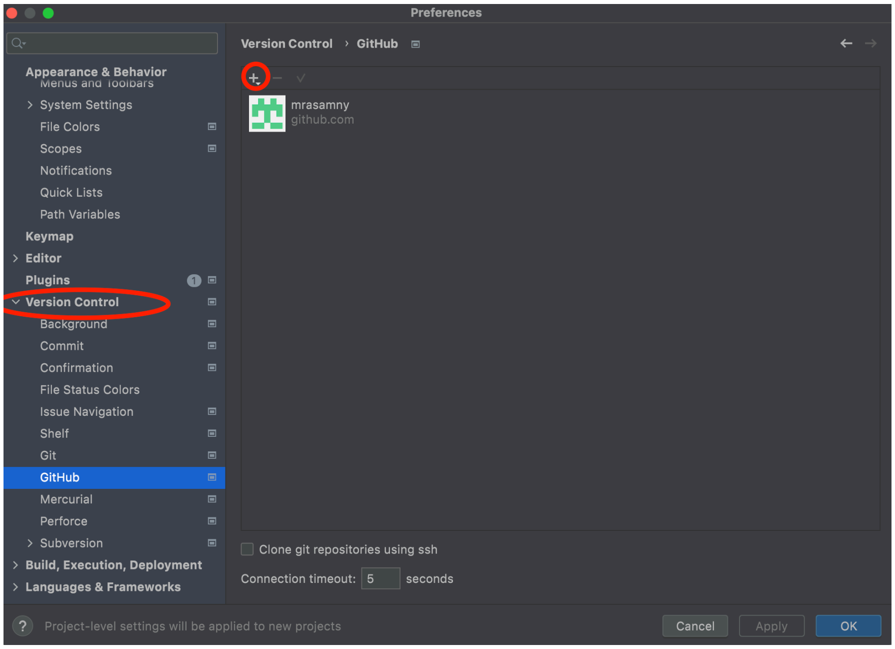
17. Expand the Version Control item on the left of the window and select Github. Then click on the + as shown in the image below.



□ Clone git repositories using ssh

Connection timeout: 5 seconds

18. Select the Login with Token and paste your token in the Token textbox and click the Add Account button as shown in the image below.

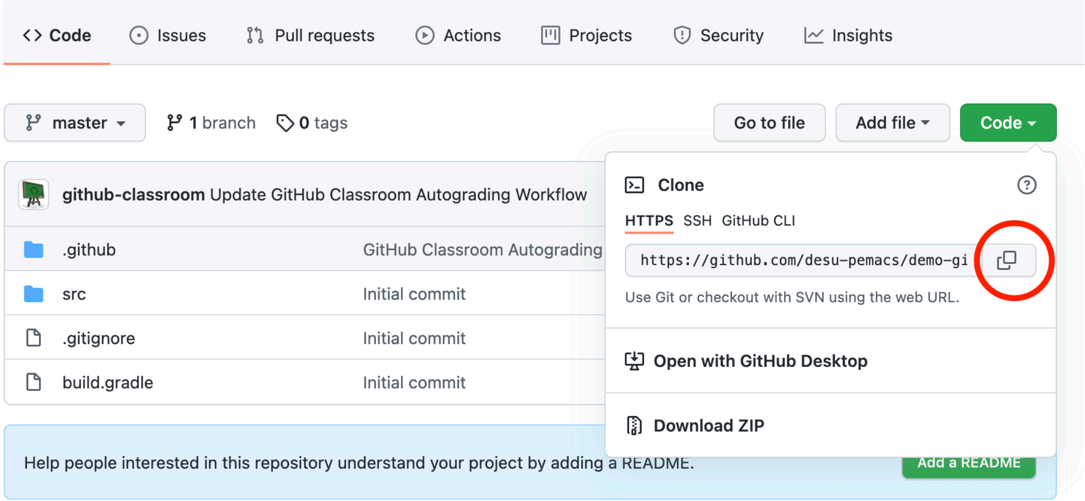


□ Clone git repositories using ssh

Connection timeout: 5 seconds

19. Now, go back to the assignment you accepted and click on the link so that you are in the repo for Demo Github Classroom.

20. Click the green code button and then copy the URL by clicking on the copy button as indicated in the image below.



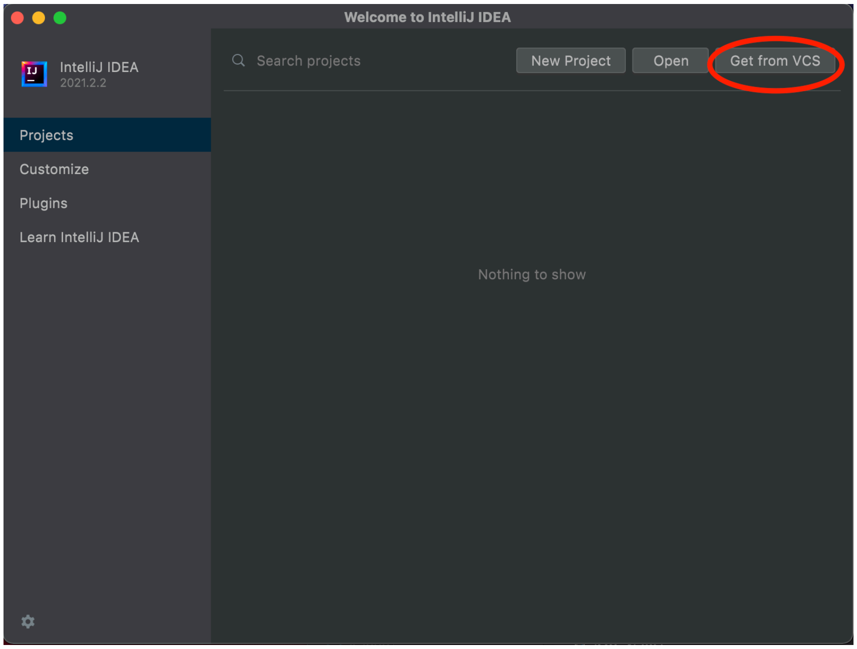
| | | |
|---|---|---|
| github-classroom Update GitHub Classroom Autograding Workflow | | |
| .github | GitHub Classroom Autograding | |
| src | Initial commit | |
| .gitignore | Initial commit | |
| build.gradle | Initial commit | |

```
https://github.com/desu-pemacs/demo-gi
```

Use Git or checkout with SVN using the web URL.

Help people interested in this repository understand your project by adding a README.

21. Open up IntelliJ and use the Get from VCS button to clone the repository.



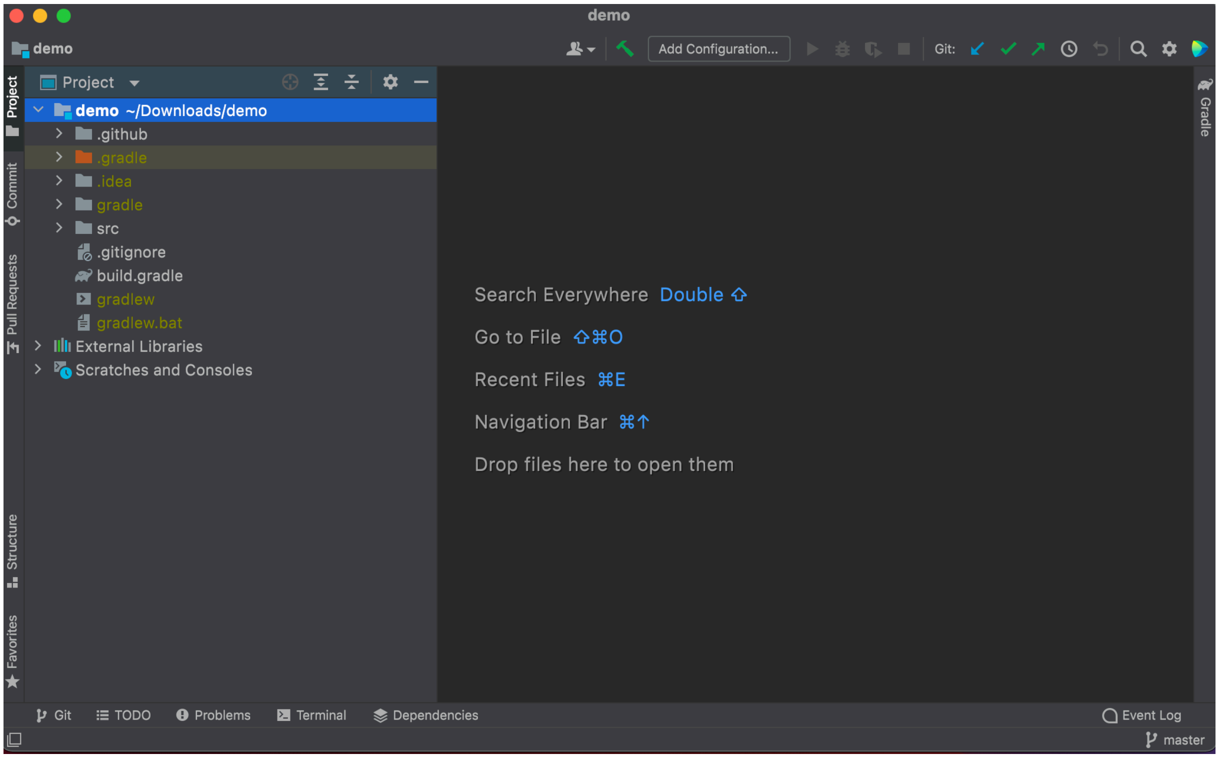
Nothing to show

22. Paste the assignment repository URL in the URL field and make sure to select the directory in which the assignment will sit. IntelliJ will add the name of repo to the Directory path. If it does not, please add the name of the assignment directory. For example, if the directory is C:\Users\JohnSmith\Documents, make sure to add the name of the assignment. In my case, this should be changed to C:\Users\JohnSmith\Documents\demo-github-classroom-rasamny

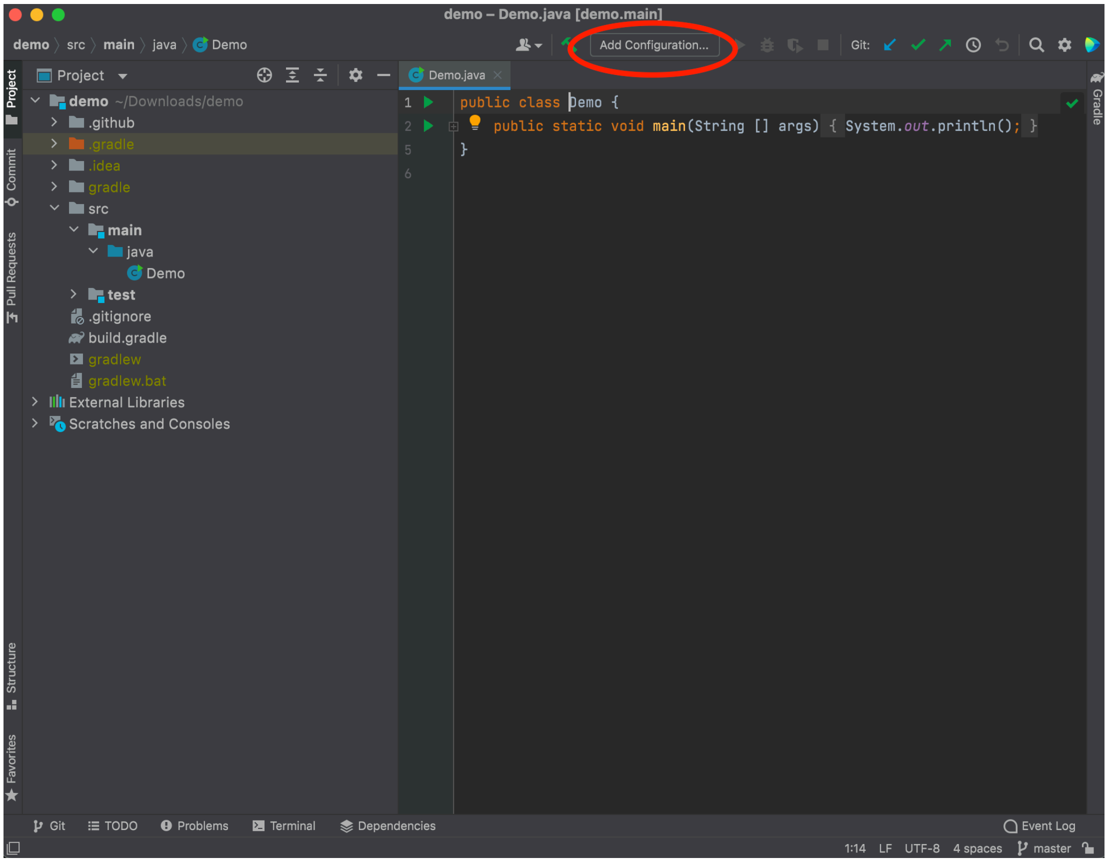
**Click the link to download and install Git, if asked.**

**WARNING:** If you do not add a folder name, IntelliJ will put all the project files in the Documents folder.

23. Press the Clone button. IntelliJ will create the folder for you and place the assignment code in the demo-github-classroom-rasamny folder under the Documents folder.
24. Your workspace should now look like the image shown below.

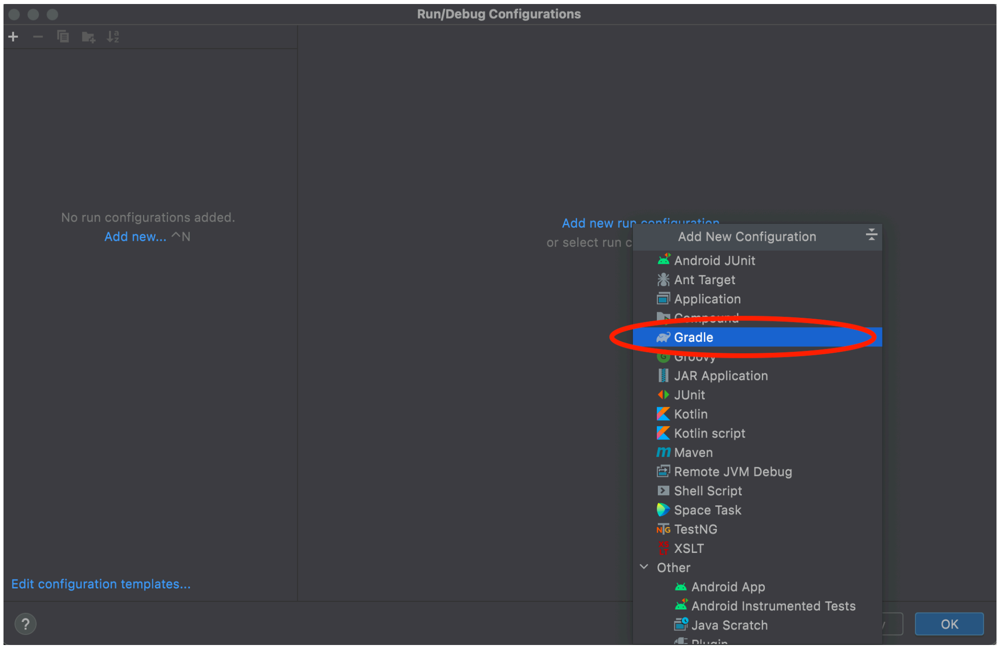


25. Now, to run and test the provided Java code, you will need to create a run configuration. Click on the run configuration button as shown in the image below.



```
public class Demo {
    public static void main(String [] args) { System.out.println(); }
}
```

26. Select Add New Configuration and then select Gradle from the menu as shown in the image below.



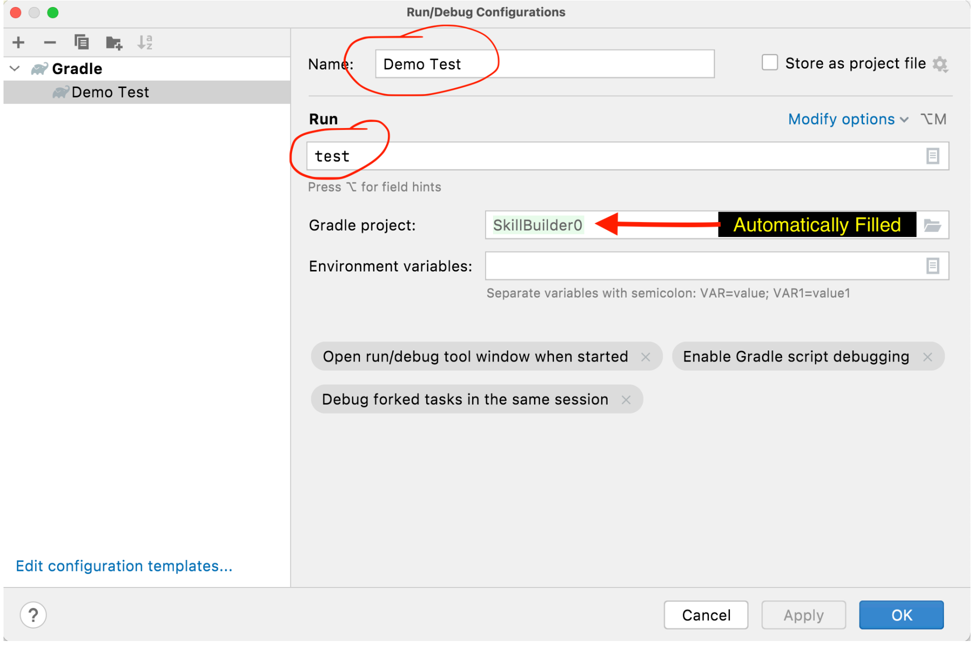
No run configurations added.

Add new... ^N

Add new run configuration

or select run c

27. Give the configuration a name, for example **Demo Test**, and type **test** in the run textbox then click OK to complete the run configuration.



Run/Debug Configurations

Gradle
- Demo Test

Name: Demo Test

□ Store as project file

Run

Modify options ⌥M

test

Press ⌥ for field hints

Gradle project: SkillBuilder0

Automatically Filled

Environment variables:

Separate variables with semicolon: VAR=value; VAR1=value1

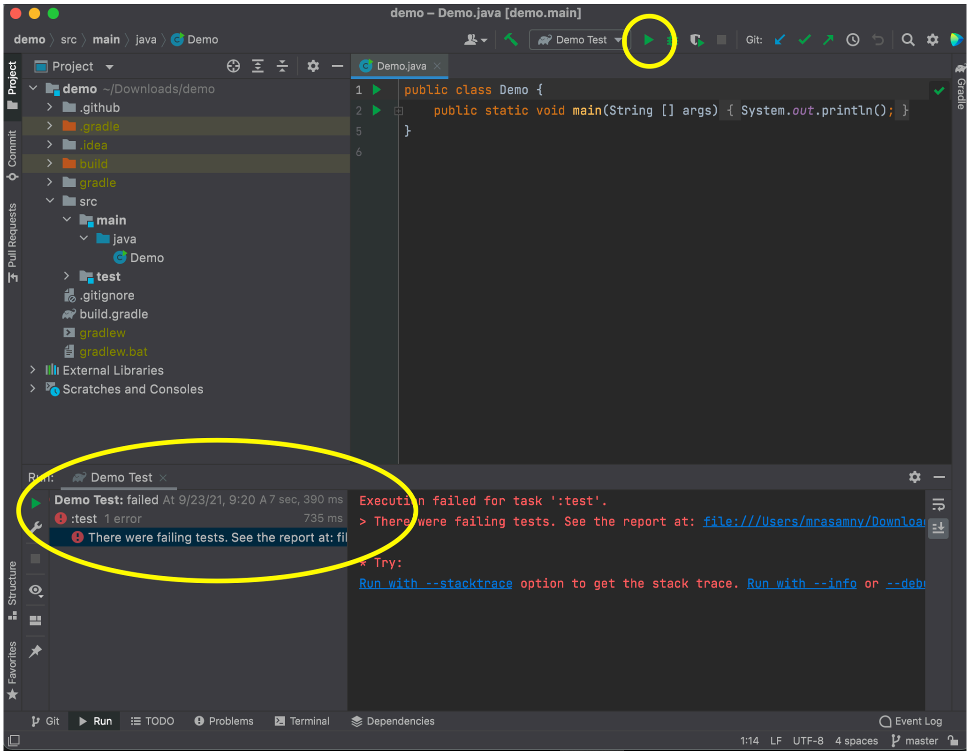
Open run/debug tool window when started

Enable Gradle script debugging

Debug forked tasks in the same session

Edit configuration templates...

28. Now press the green play button and notice that the test will fail.



demo – Demo.java [demo.main]

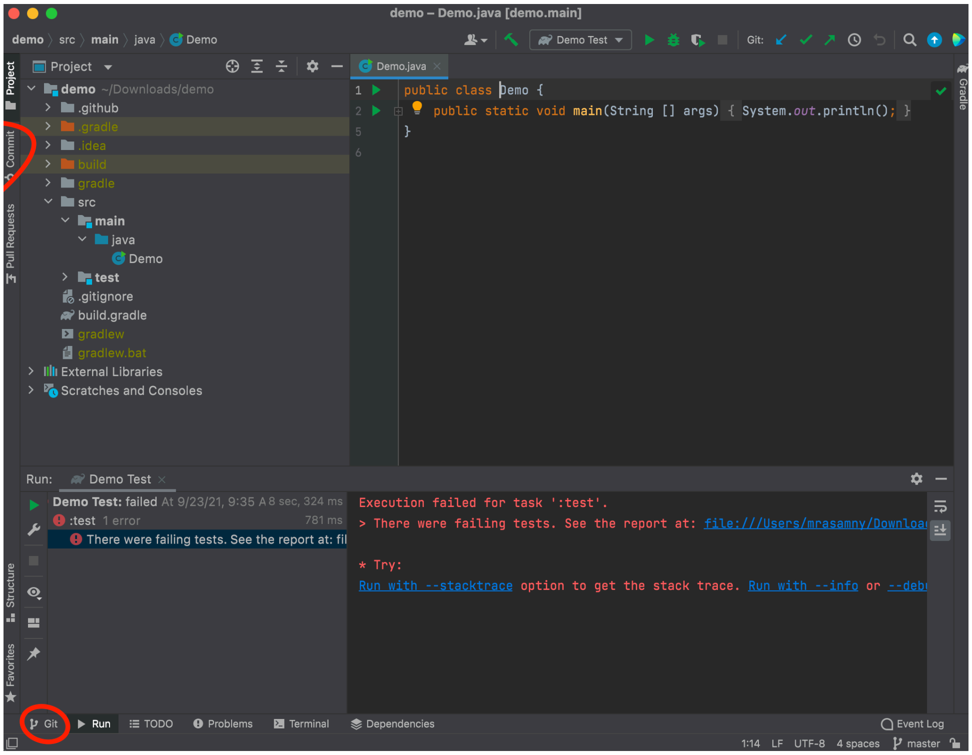
```
public class Demo {
    public static void main(String [] args) { System.out.println(); }
}
```

Demo Test: failed At 9/23/21, 9:20 A 7 sec, 390 ms
:test 1 error 735 ms
There were failing tests. See the report at: fil

```
Execution failed for task ':test'.
> There were failing tests. See the report at: file:///Users/mrasamny/Downloa

* Try:
Run with --stacktrace option to get the stack trace. Run with --info or --debu
```

29. Add “Hello World!” so that the following line of code reads `System.out.println("Hello World");` and then run the test again by clicking on the green play button. You should now see that the test suucceeds!
30. You now need to commit the changes you made and push the changes to your repository on Github. Select the Commit tab as shown in the image below.



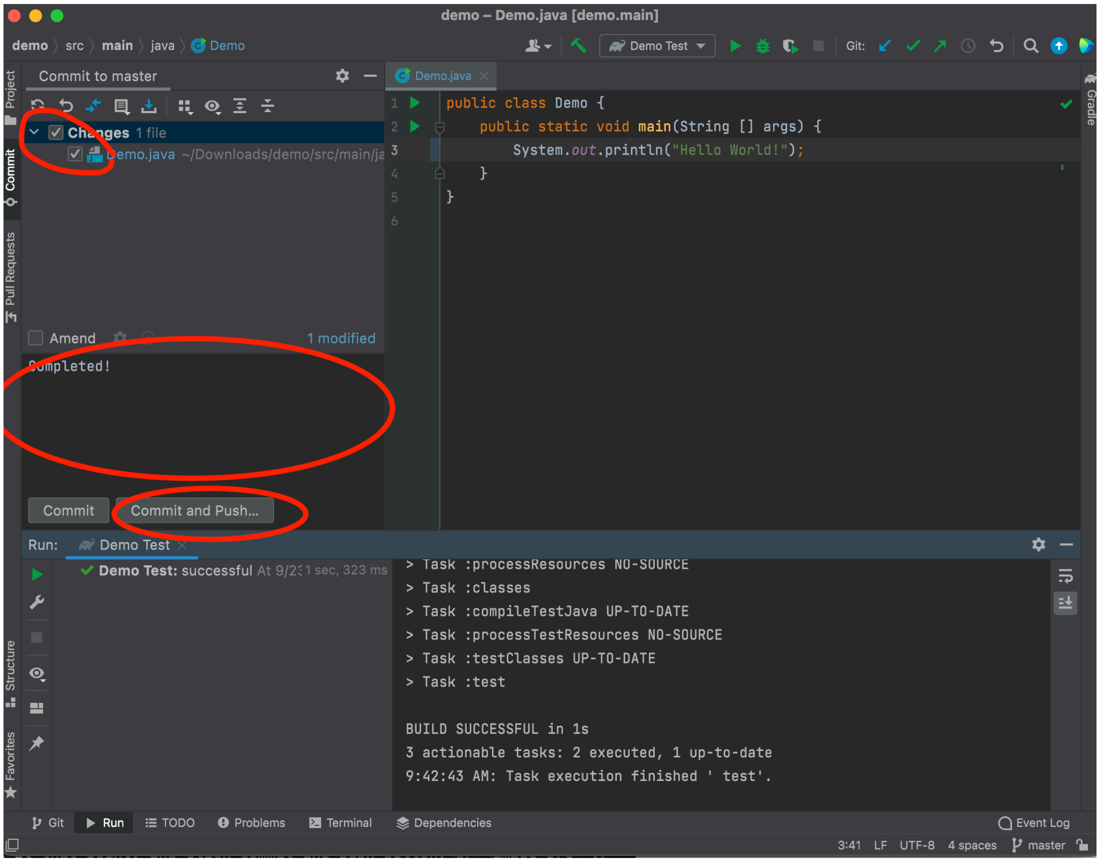
```
public class Demo {
    public static void main(String [] args) { System.out.println(); }
}
```

```
Demo Test: failed At 9/23/21, 9:35 A 8 sec, 324 ms
:test 1 error 781 ms
There were failing tests. See the report at: fil
```

```
Execution failed for task ':test'.
> There were failing tests. See the report at: file:///Users/mrasamny/Downloa

* Try:
Run with --stacktrace option to get the stack trace. Run with --info or --debu
```

31. Select all the files that have changed and type a message in th text box as shown in the image, the message should summarize the work done at that point, and click commit and push button.



```
public class Demo {
    public static void main(String [] args) {
        System.out.println("Hello World!");
    }
}
```

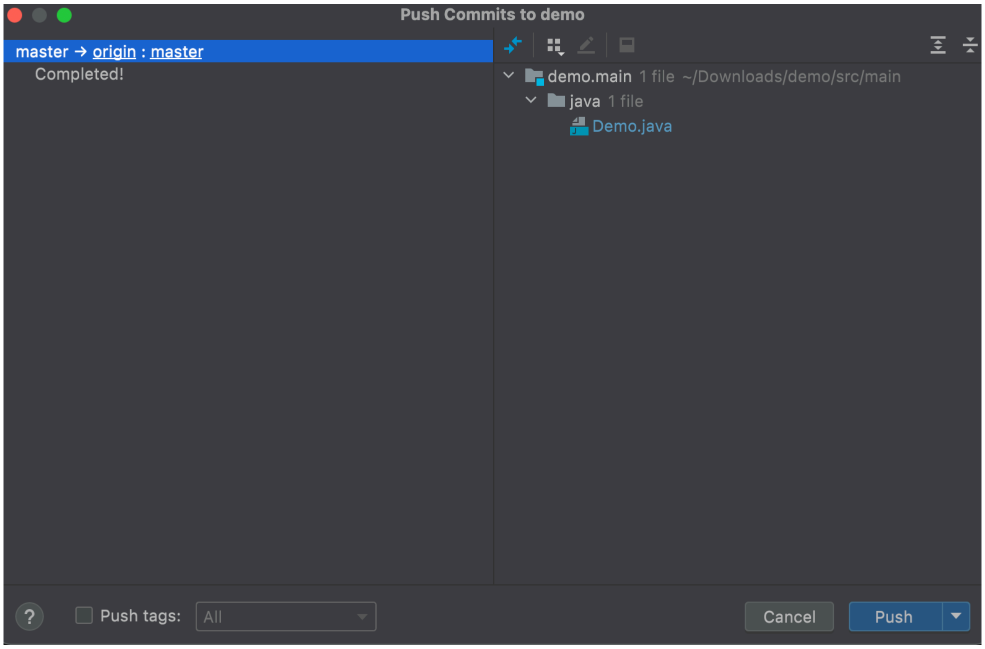
```
Completed!
```

```
Demo Test: successful At 9/2: 1 sec, 323 ms
```

```
> Task :processResources NO-SOURCE
> Task :classes
> Task :compileTestJava UP-TO-DATE
> Task :processTestResources NO-SOURCE
> Task :testClasses UP-TO-DATE
> Task :test

BUILD SUCCESSFUL in 1s
3 actionable tasks: 2 executed, 1 up-to-date
9:42:43 AM: Task execution finished ' test'.
```

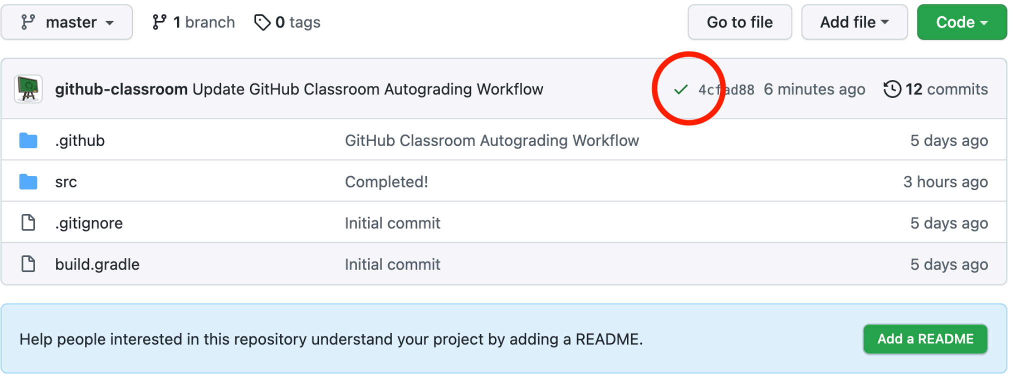
32. Click on the push button in the next window to complete the submission.



master → origin : master
Completed!

demo.main 1 file ~/Downloads/demo/src/main
java 1 file
Demo.java

33. Go back to your repo on Github and make sure that you see a check mark indicating that everything passed. See image below.



github-classroom Update GitHub Classroom Autograding Workflow ✓ 4cfad88 6 minutes ago 12 commits

| | | |
|---|---|---|
| .github | GitHub Classroom Autograding Workflow | 5 days ago |
| src | Completed! | 3 hours ago |
| .gitignore | Initial commit | 5 days ago |
| build.gradle | Initial commit | 5 days ago |

Help people interested in this repository understand your project by adding a README.

**Congratulations! You now have completed your first Java assignment!**