GitHub Collaboration

Welcome to the New Collaboration!

Git What?

- If you have not done so already, review the descriptions, materials, and video detailed in the Introduction to Git portion of section 1.8 Git and GitHub on Blackboard.
- You will gain an understanding of Git, version control, and why it’s important.

More Background

- Review Chapter 1 Getting Started
- Review Section 1.1 About Version Control
- Review Section 1.2 A Short History of Git
- Review Section 1.3 Git Basics
- You can navigate these sections using the “Prev” and “Next” links at the bottom of the page.

Recapping 1a

- You installed and configured Git on your local machine.
- You created a JointProject folder and launched the Git processes and code for that folder.
- You created a file called README in the folder.
- You tracked the README file with “git add”, created a snapshot of the project using “git commit”, and commented on your commit.
- You checked your progress and commit comment using “git log”.

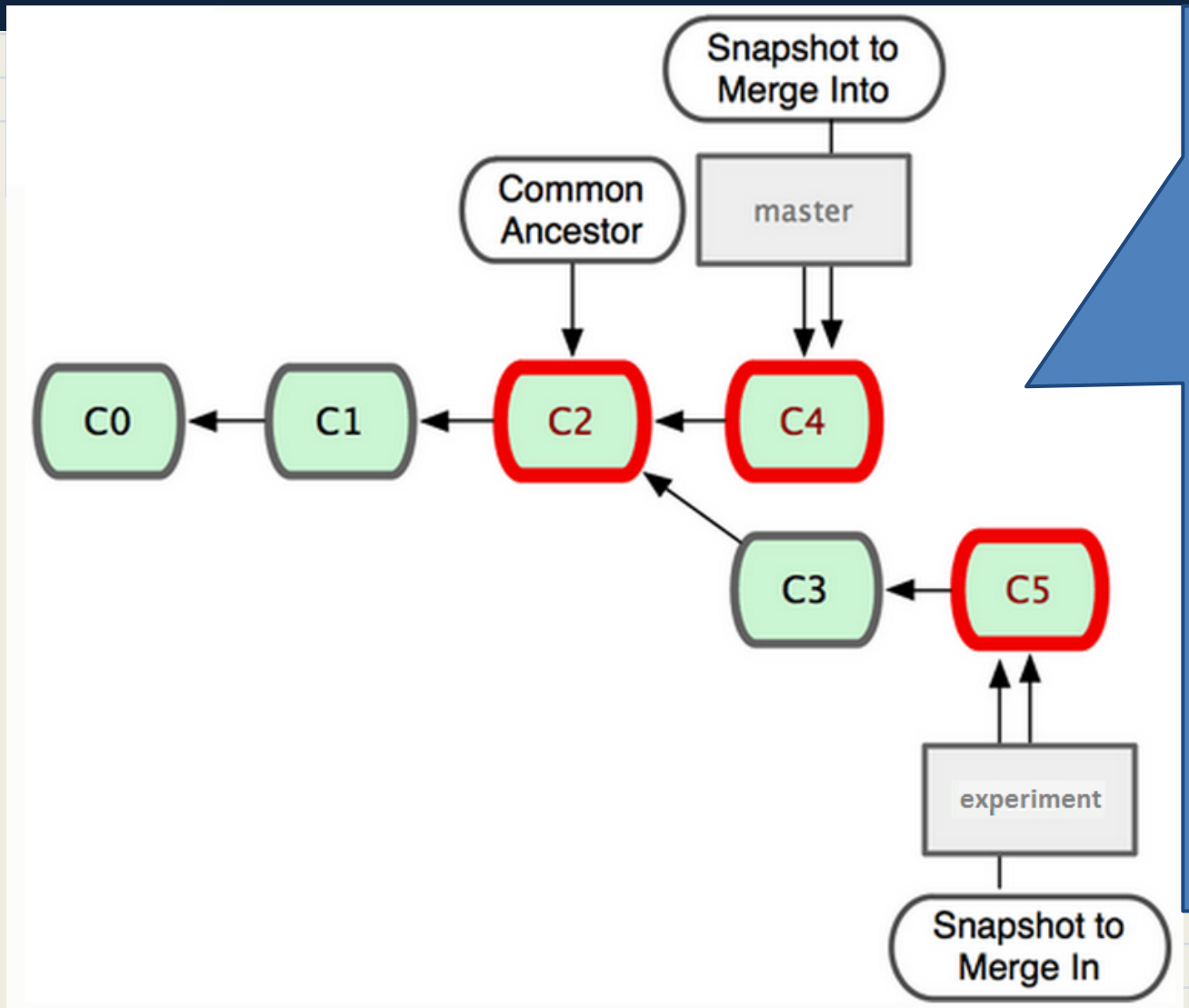
Recapping 1b

- You explored the master branch making changes to your files and committed them, creating new snapshots (C1-C2).
- You created a new branch (experiment), made changes in files and committed them, creating a separate project snapshot (C3).
- You created a new branch (surething), made changes in files and committed them, creating snapshots separate from both the master and experiment branches (C4).

Recapping 1b

- You merged surething and master. No “merge conflict” existed because master was a “direct ancestor” of surething.
- You returned to the experiment branch and made a few more changes and commits (C5).
- Attempting to merge experiment and master, you discovered a “merge conflict” arose.
- By editing the conflicted file (READMETOO) and committing it, you resolved the conflict and created a new inclusive snapshot (C6).

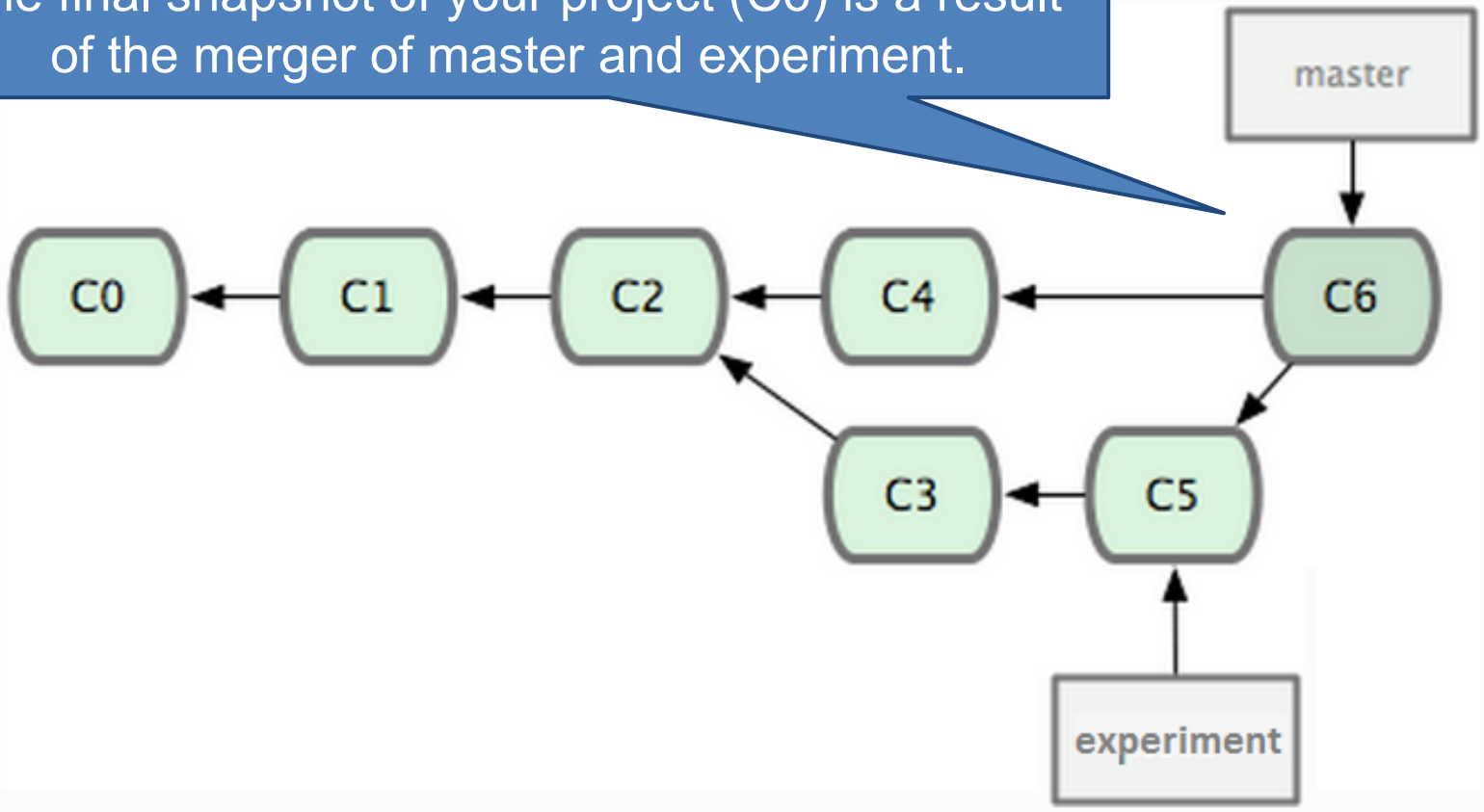
JointProject C5 Merge Plan



Master is NOT a direct ancestor of experiment, so Git can't just fast forward the master branch to a new merged project snapshot. Git finds the closest common ancestor of the two states and brings all three together.

JointProject C6

The final snapshot of your project (C6) is a result of the merger of master and experiment.



Recapping 1c

- You explored the relationship between Git (a version control system) and GitHub (a repository and social work website for Git).
- You created a GitHub account and a repository.
- You created a Hello-World project with a new README file, then staged, committed and commented on it.
- You set up a pointer from your local machine to GitHub using “git remote”.

Recapping 1c

- Using the remote pointer you created, you pushed your new project (Hello-World) to GitHub.
- You checked your work by returning to your GitHub repository and seeing the new README file posted there.
- You listed your GitHub username and GitHub Joint Project repository name in the GitHub Directory file in Jurkat Fall 2013 Google Drive folder.

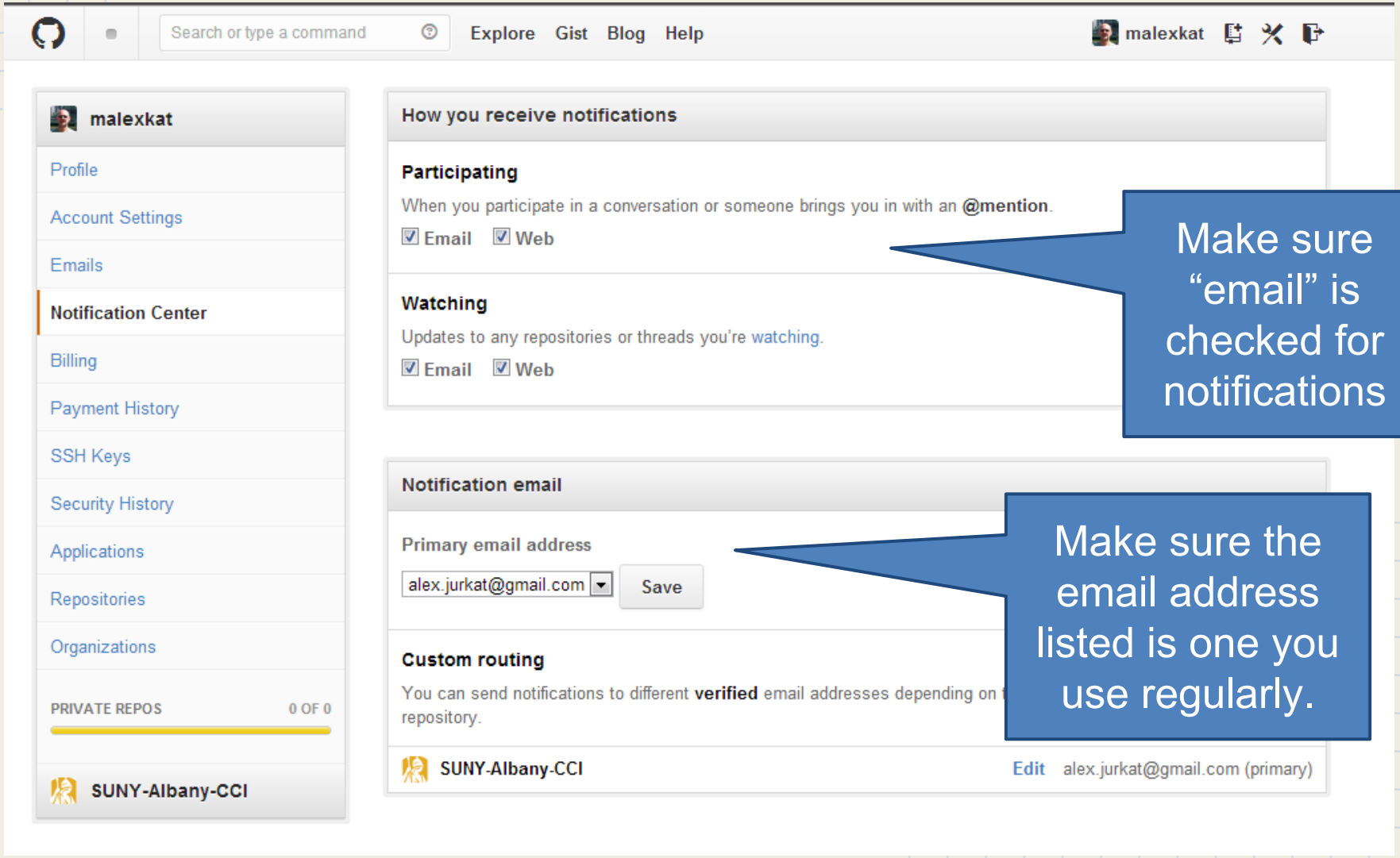
Putting the “Joint” in Joint Project

Let’s get started!

Checking Communications

- This exercise will work better if your communications lines from GitHub are open.
- Browse to GitHub and sign into your account.
- Check https://github.com/settings/notifications.
- Make sure the email address listed on that page is one you check regularly.

Checking Communications



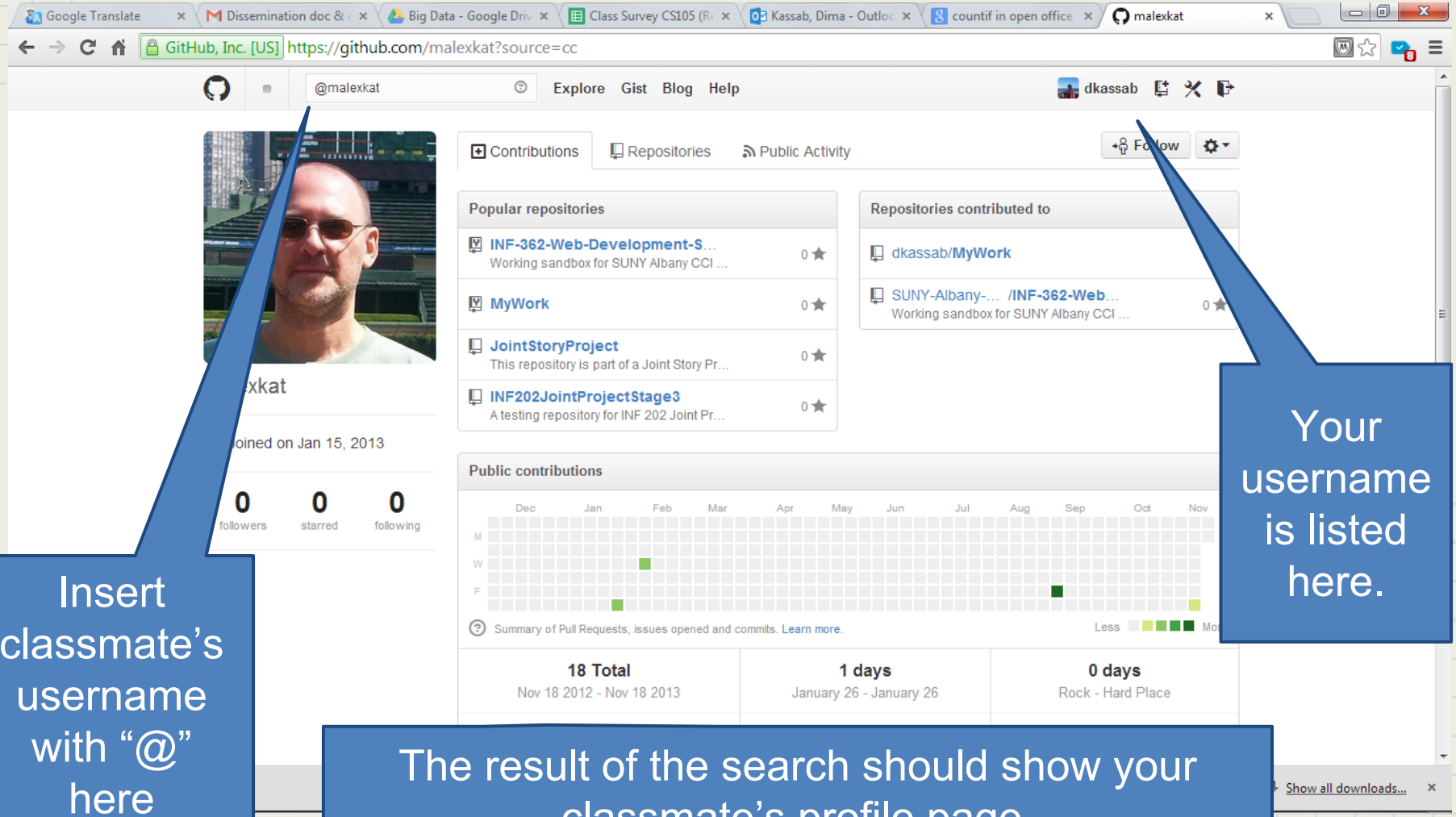
Finding a Project

- Review the GitHub Assignments spreadsheet in the Git & GitHub directory of the Google Drive Jurkat Fall 2013 folder.
- Find a classmate with a blank cell in the Contributor 1, Contributor 2, or Contributor 3 field. If a lower number contributor cell is open (such as Contributor 1), favor it over a higher number contributor cell. Type your name in one of those blank cells.
- Note down that classmate's GitHub username and repository name.

Finding a Project

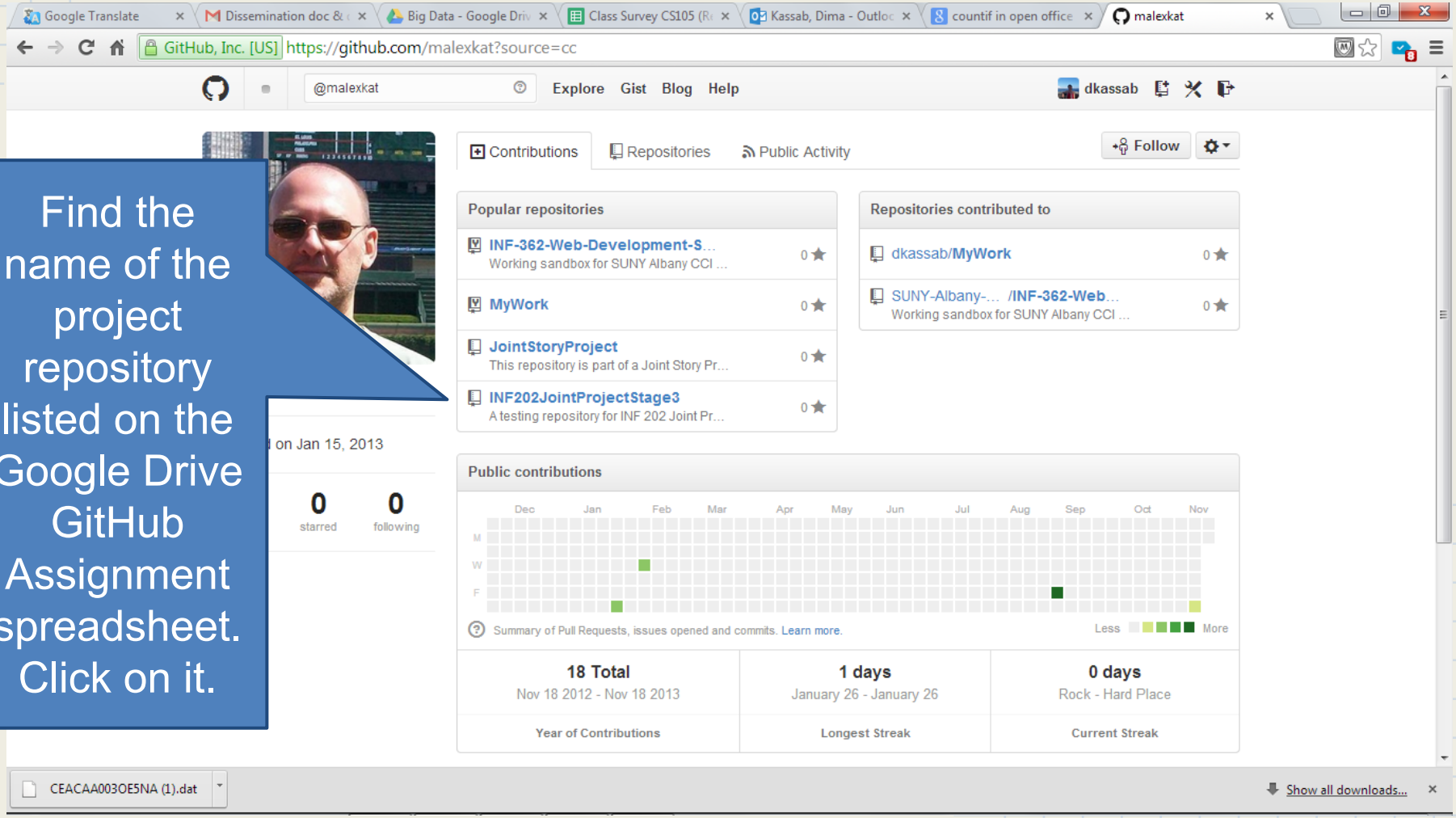
- Browse to GitHub.com.
- Sign in (if necessary).
- Type your classmate's username in the search box at the top of the website, preceded by an "@" symbol (no quotes).
- You should be directed to that classmate's profile page.
- In the example slides that follow (16-36), Dima Kassab (@dkassab) contributes to Alex Jurkat's project.

Classmate's Profile Page

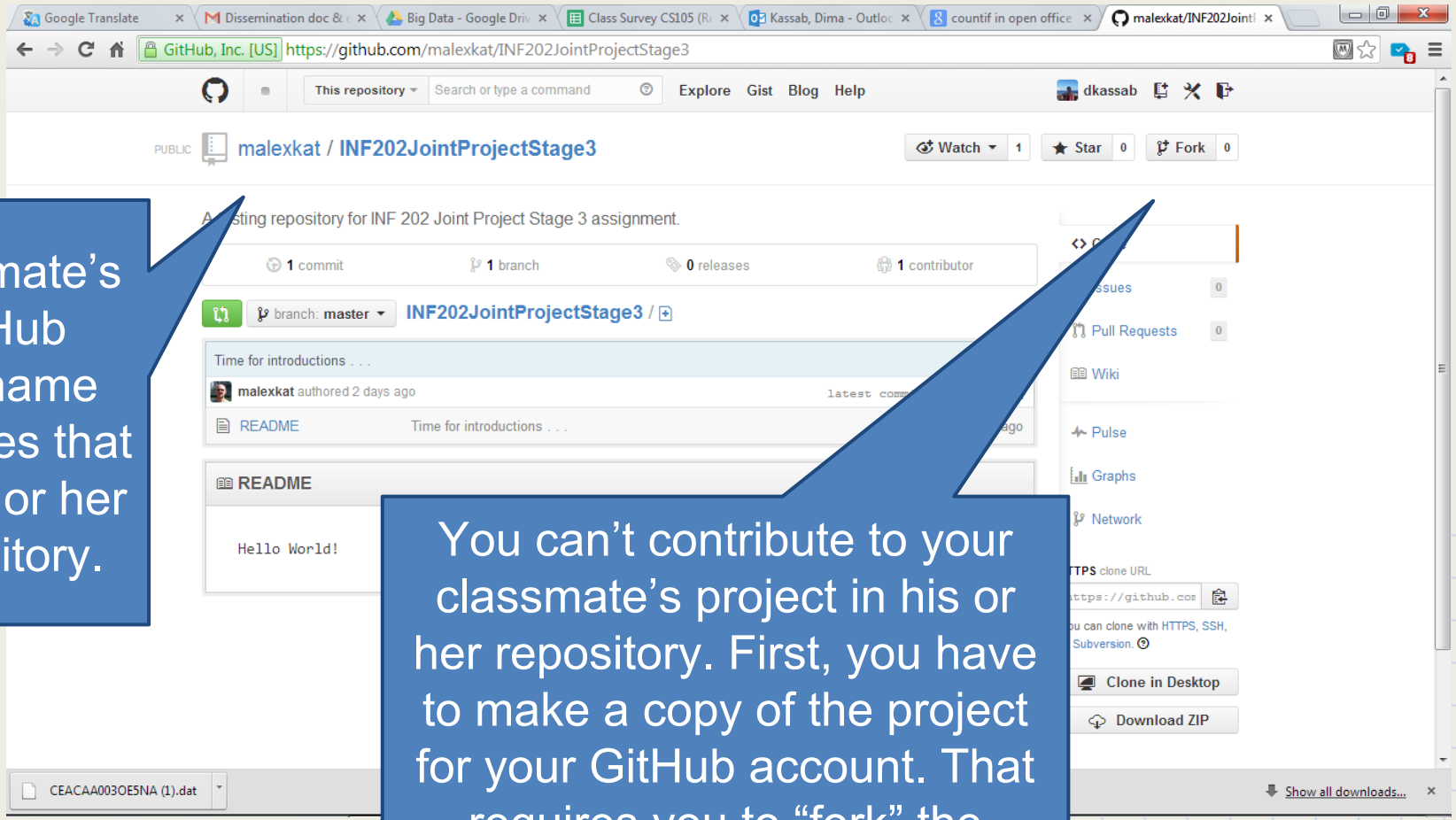


Classmate's Profile Page

Find the name of the project repository listed on the Google Drive GitHub Assignment spreadsheet. Click on it.



Classmate's Repository

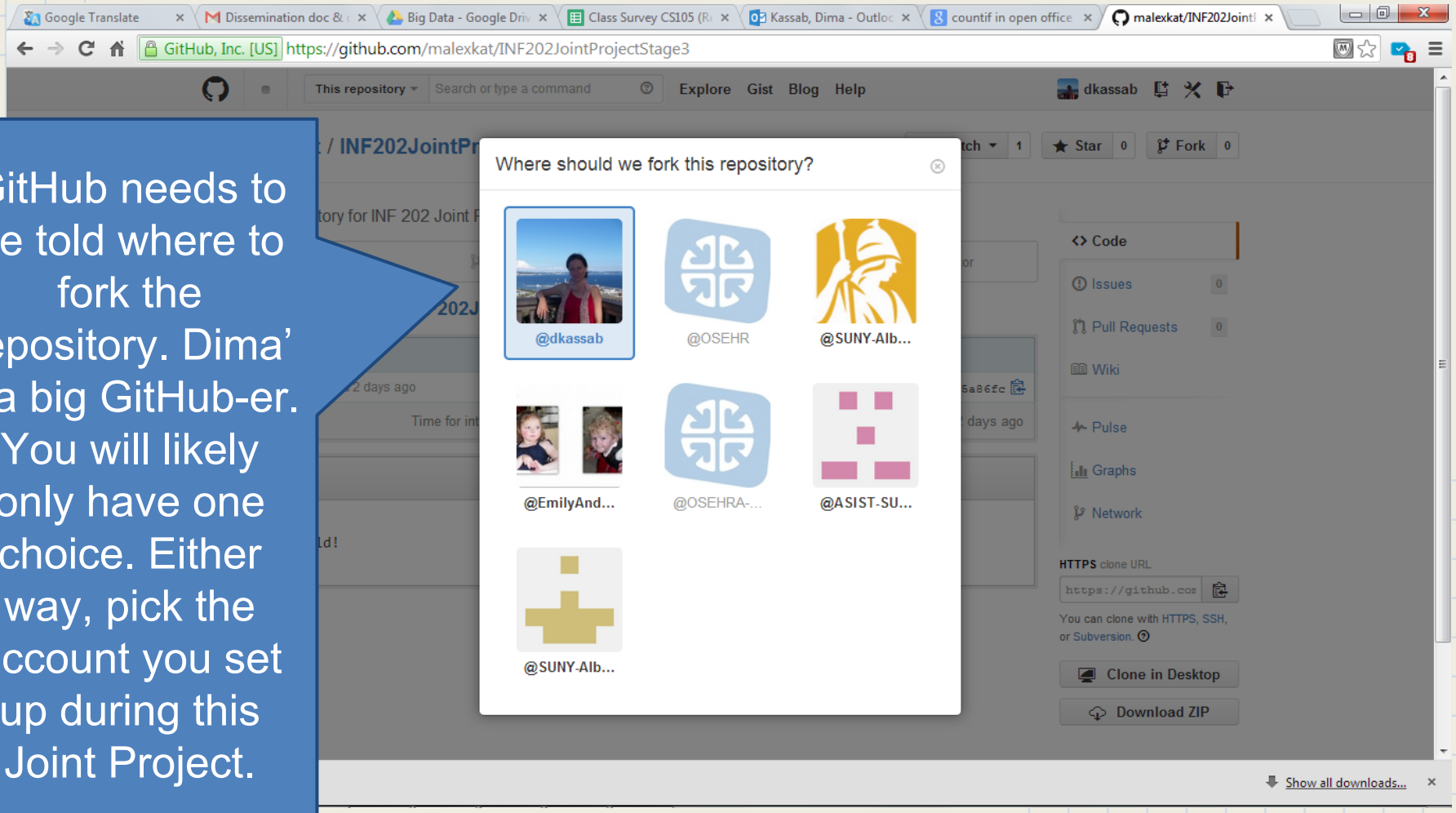


Classmate's GitHub username indicates that it's his or her repository.

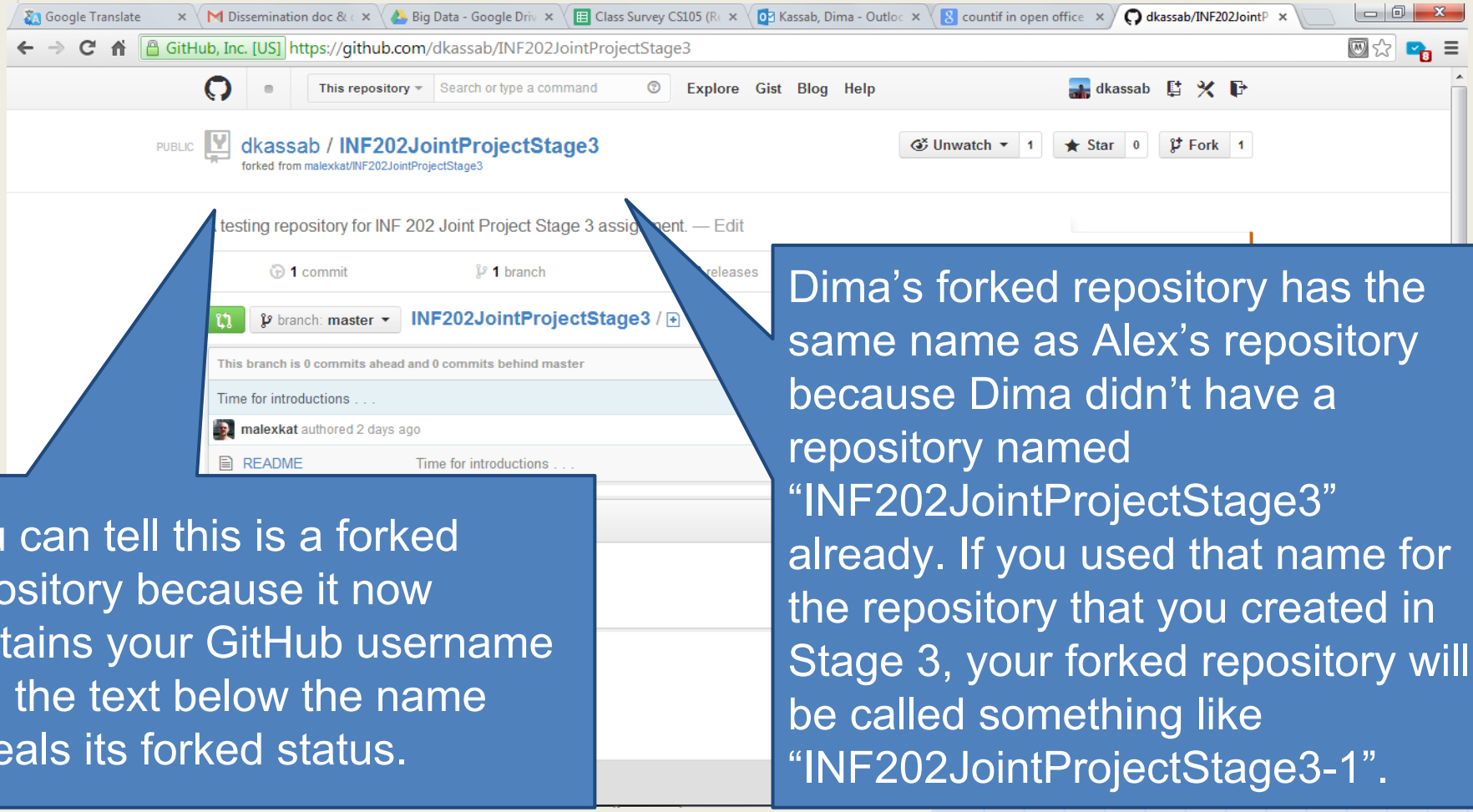
You can't contribute to your classmate's project in his or her repository. First, you have to make a copy of the project for your GitHub account. That requires you to "fork" the repository. Click here.

Whither the Fork?

GitHub needs to be told where to fork the repository. Dima' s a big GitHub-er. You will likely only have one choice. Either way, pick the account you set up during this Joint Project.



Whither the Fork?



You can tell this is a forked repository because it now contains your GitHub username and the text below the name reveals its forked status.

Dima's forked repository has the same name as Alex's repository because Dima didn't have a repository named "INF202JointProjectStage3" already. If you used that name for the repository that you created in Stage 3, your forked repository will be called something like "INF202JointProjectStage3-1".

Back to Git and Local Machine

- You have a forked repository on GitHub, but you still can't contribute to your classmate's project.
- You need to bring the project to your local machine to manipulate the files.
- Click away from your browser and open Git Bash.
- Use ls, pwd, and cd to move into the JointProject folder you created for the prior stages. (If you have trouble, see Assignment 1b slides.)

Make a Space

Create a subfolder in your JointProject folder. Make sure your classmate's name is in the folder name you create.

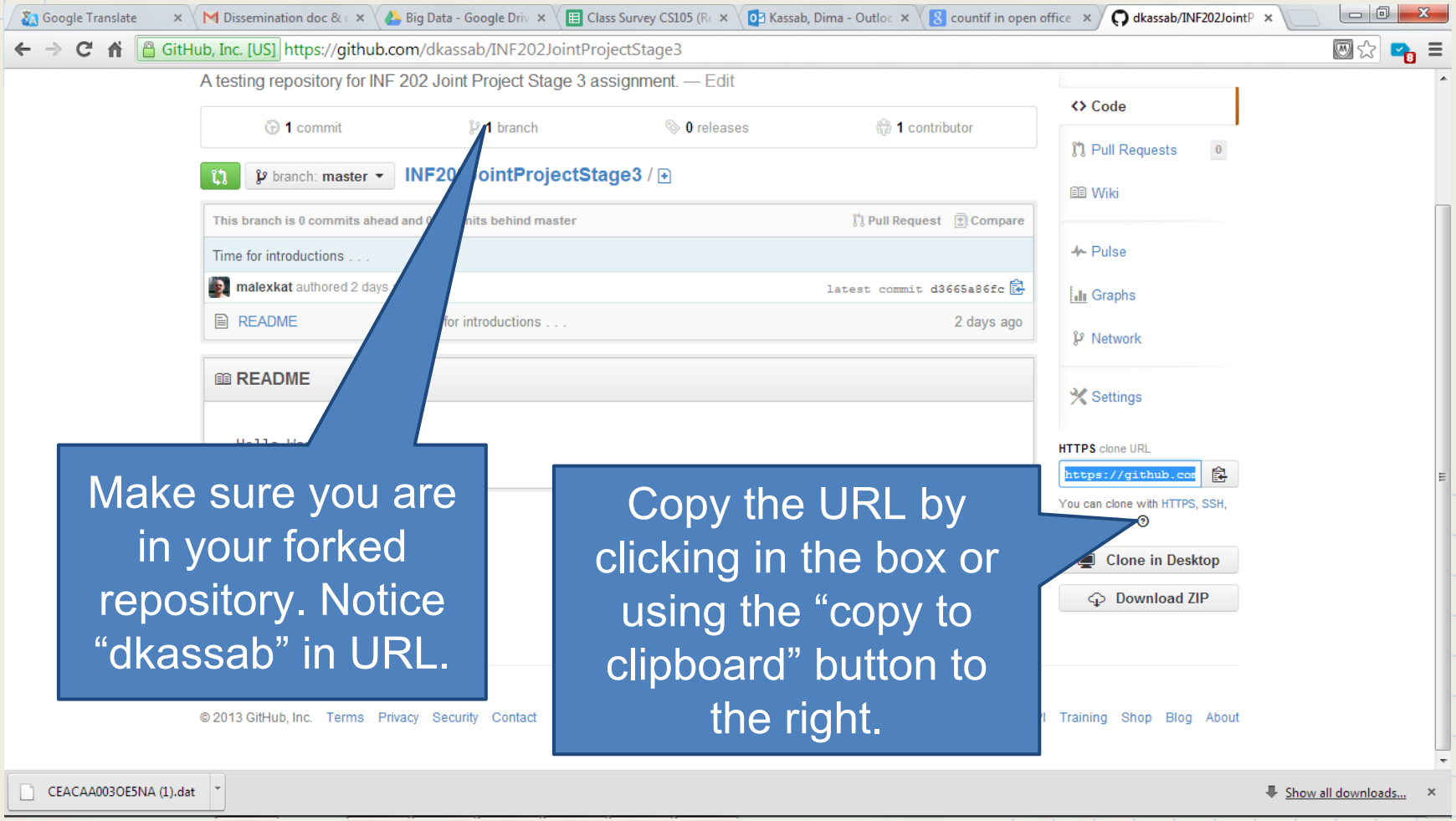
Move into that new directory.

If you like, git-ify that new directory. This step is not entirely necessary, however. When you clone the repository in slides 25-26, the folder becomes git-ified automatically.

Finding the Clone URL

- Now that you have a space for your work, it's time to clone the repository from GitHub to your local machine.
- Browse back to GitHub and open the forked repository.
- Look to the bottom right for the HTTPS clone URL text box.
- Copy the URL from that box into Notepad or Word or some other readable file.

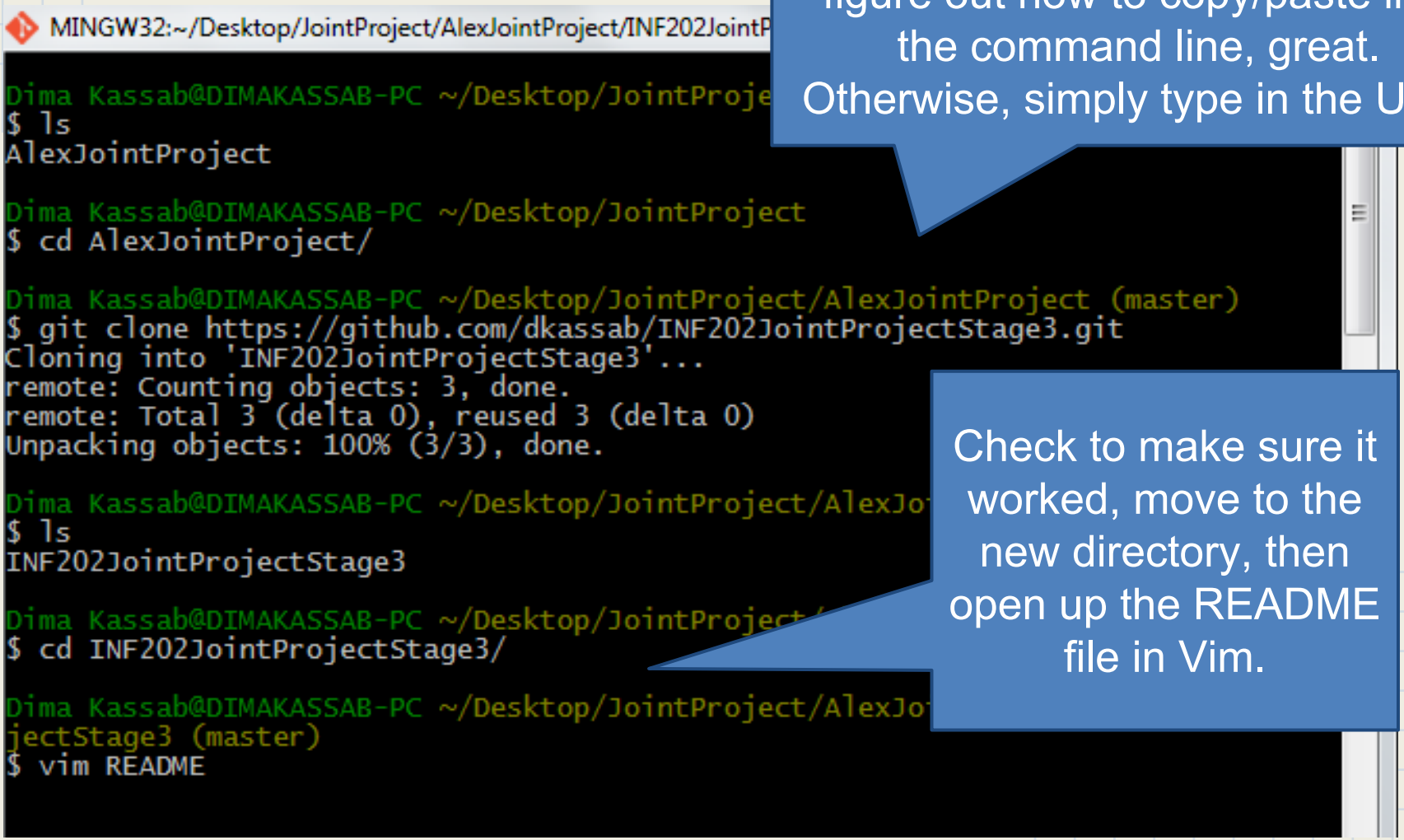
Finding the Clone URL



Cloning to Local Machine

- Return to Git Bash.
- Make sure you are in the new directory that you created for your classmate's project.
- Clone your GitHub forked repository to your local machine using "git clone" and the URL you got from GitHub in slide 24.
- Make sure the new folder and file is there, then open up the README file using Vim.

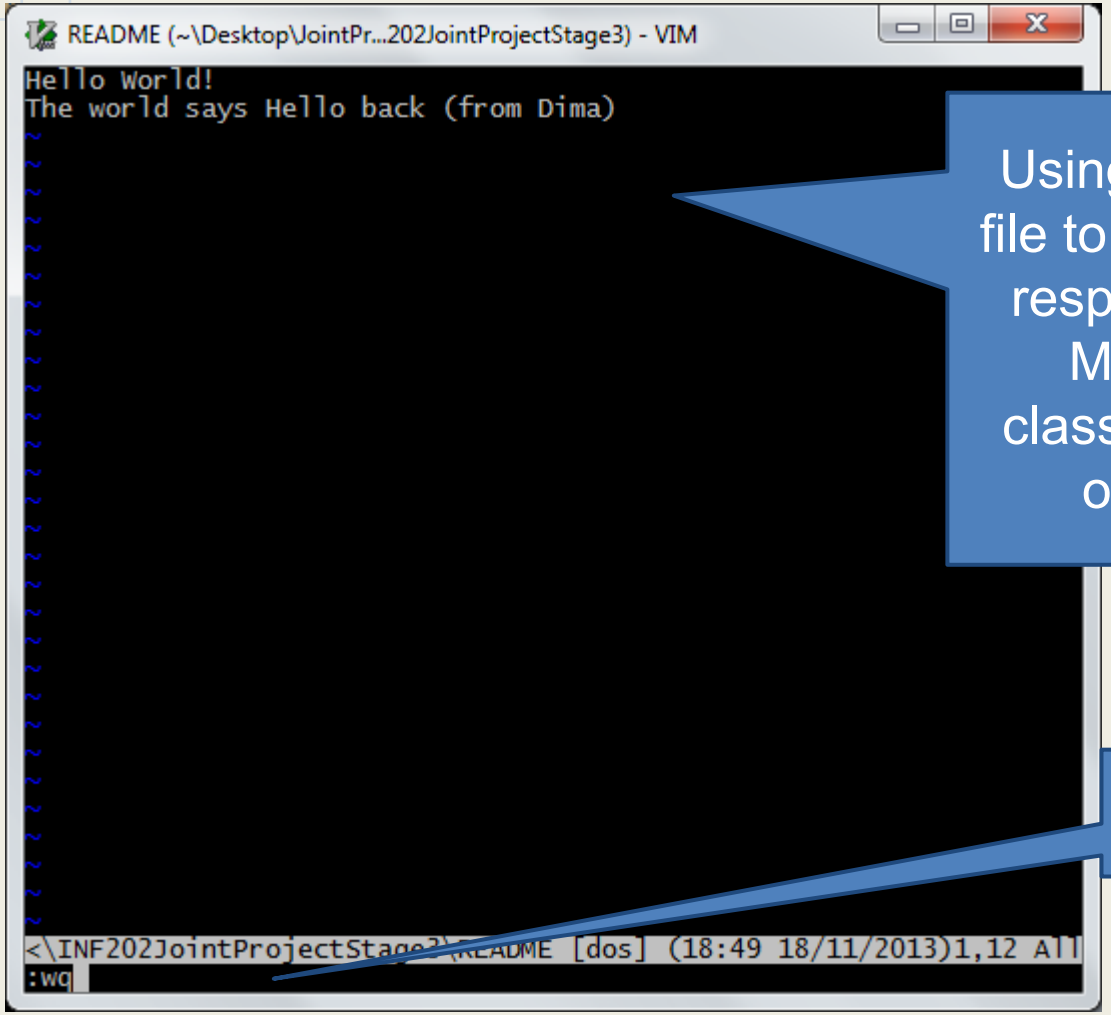
Cloning to Local



After making sure you are in the right directory, clone the forked repository using its URL. If you can figure out how to copy/paste into the command line, great. Otherwise, simply type in the URL.

Check to make sure it worked, move to the new directory, then open up the README file in Vim.

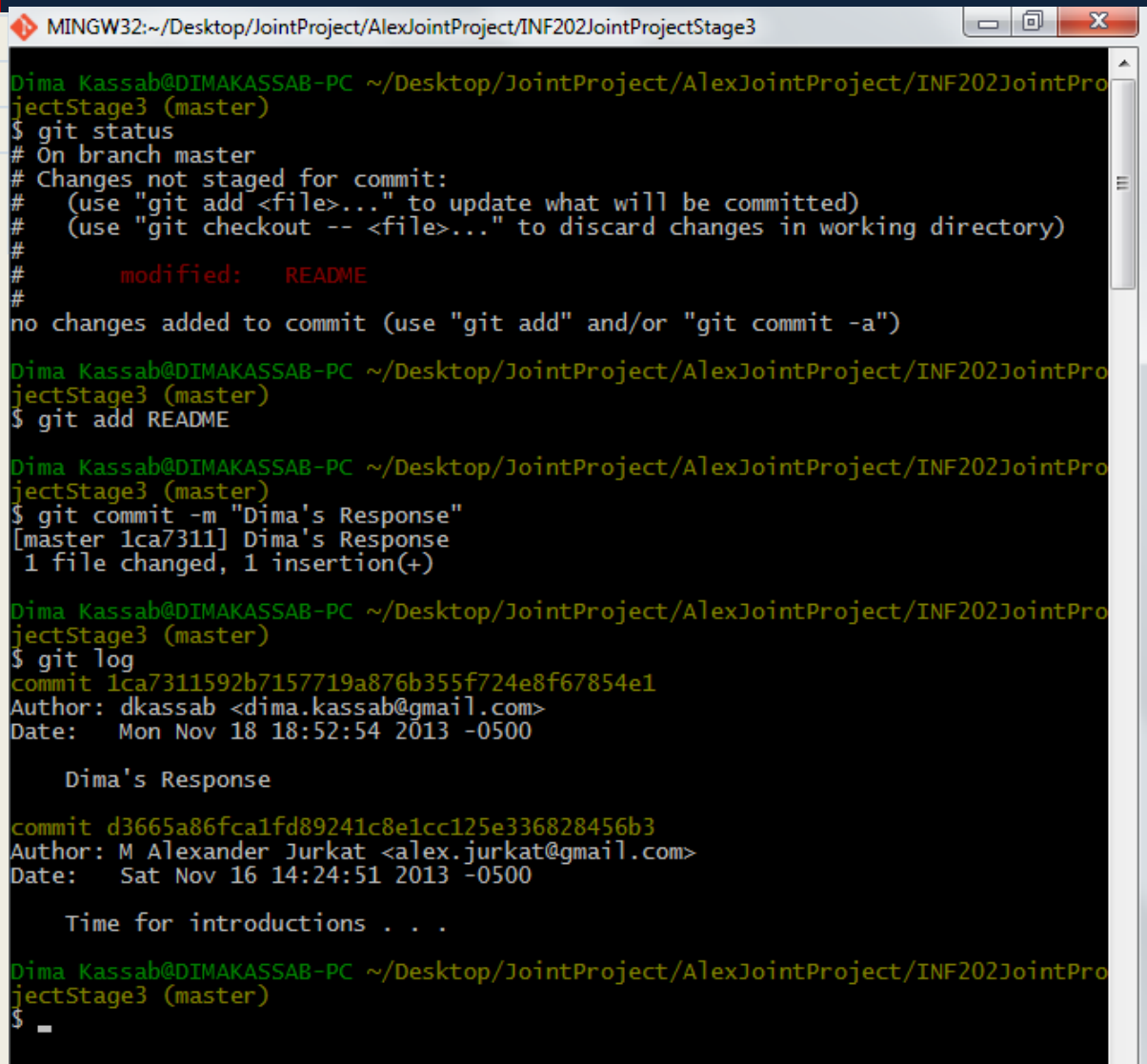
Speaking for the World



Using Vim, edit the README file to add your own welcoming response to your classmate. Make sure you tell your classmate who is responding on behalf of the world.

Save your changes.

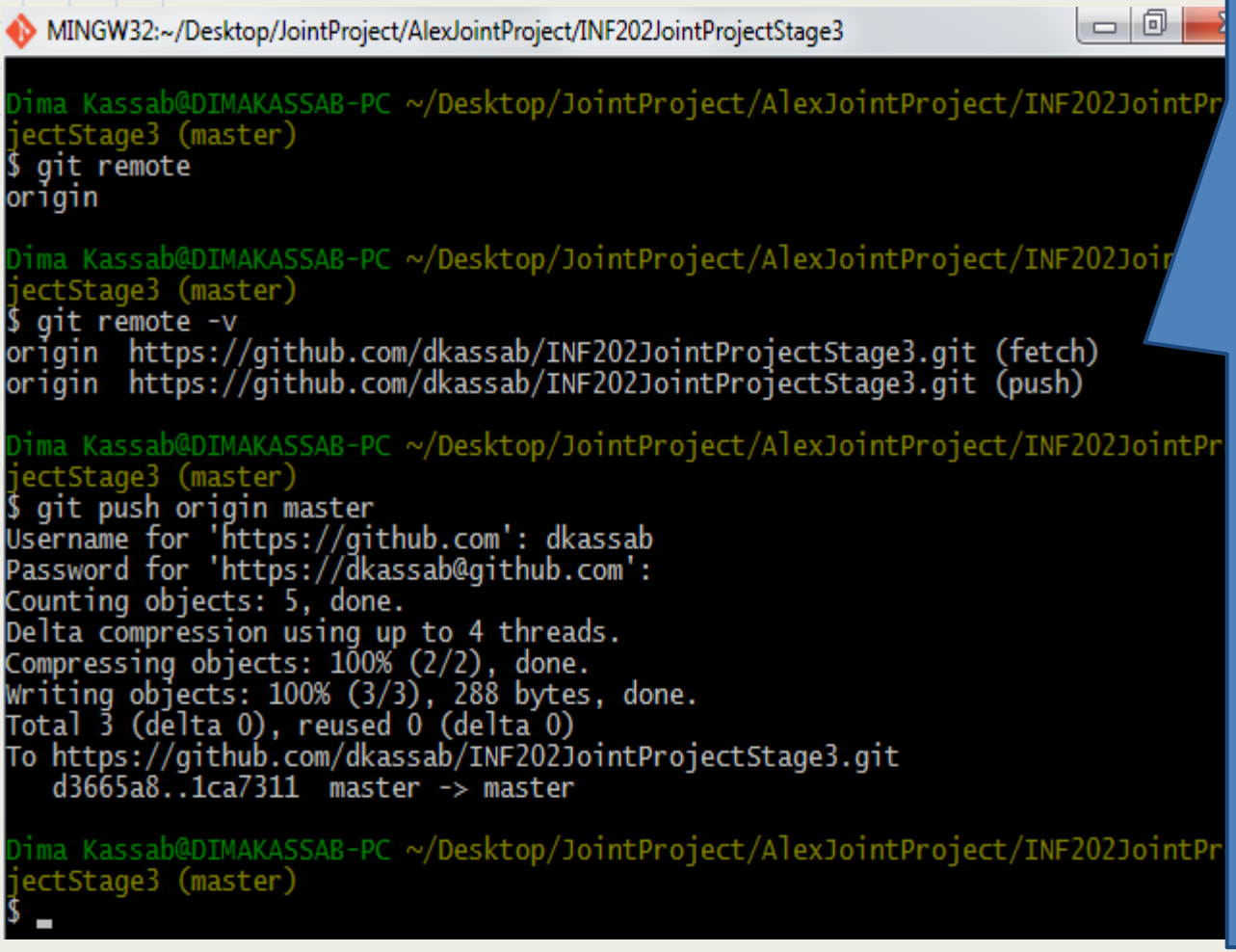
Stage, Commit, Comment



Stage, commit, and comment your changes, then check the log. This process should be fairly familiar by now. The example here shows the “shortform” commit/ comment process (using “-m”). Make sure you replace “Dima’s Response” with your own comment.

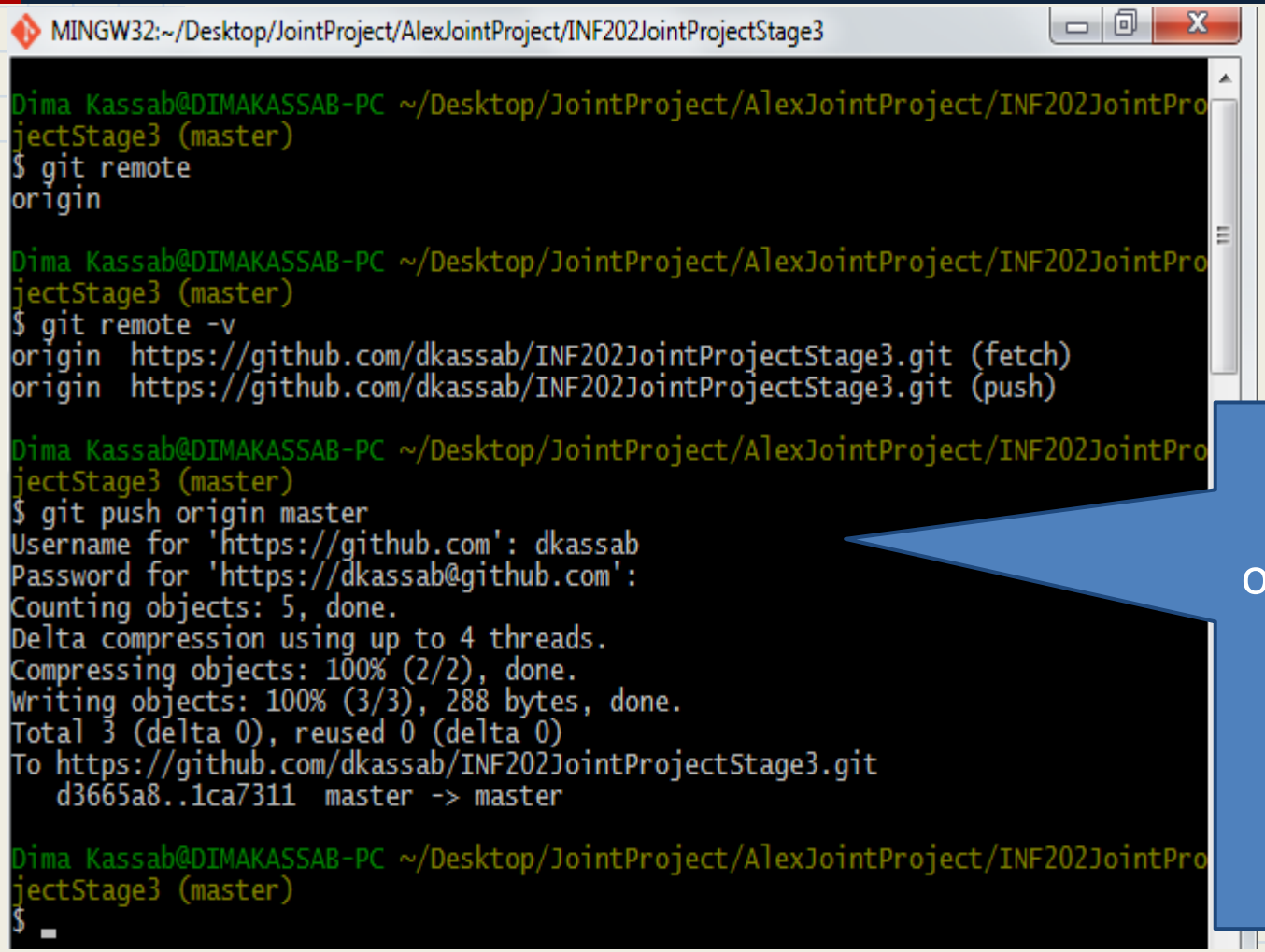
Remember this is all being done on your local machine.

Push Changes to Forked Repository



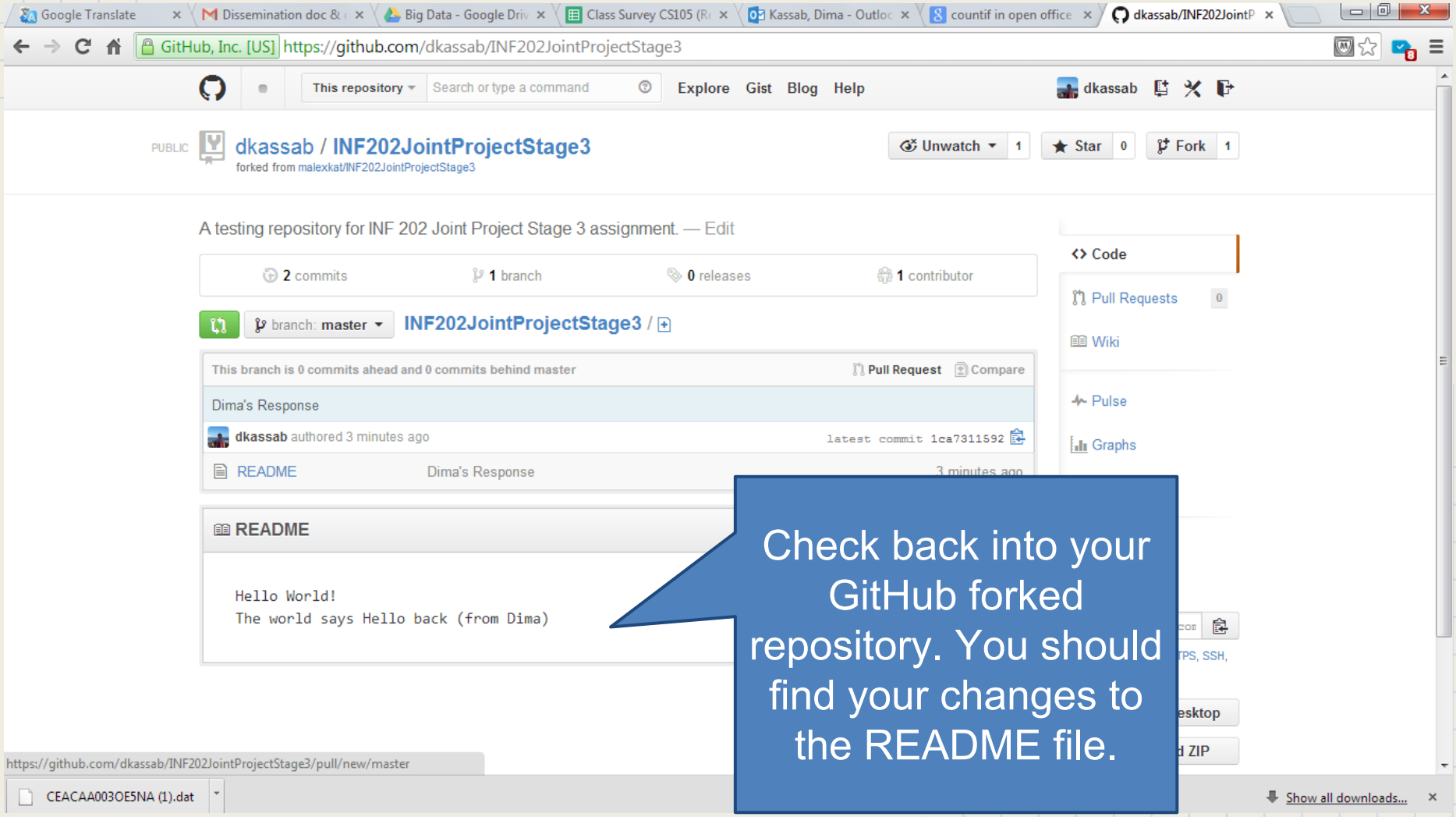
In Assignment 1c, you created a remote pointer to a GitHub repository. In this stage, Git automatically created a similar remote pointer and called it “origin” when you cloned the forked repository in slide 26. Check by typing “git remote”. You should find the “origin” pointer. See more details using “git remote -v”.

Push Changes to Forked Repository



Push your changes on this project to your forked repository using the “git push origin master” command.

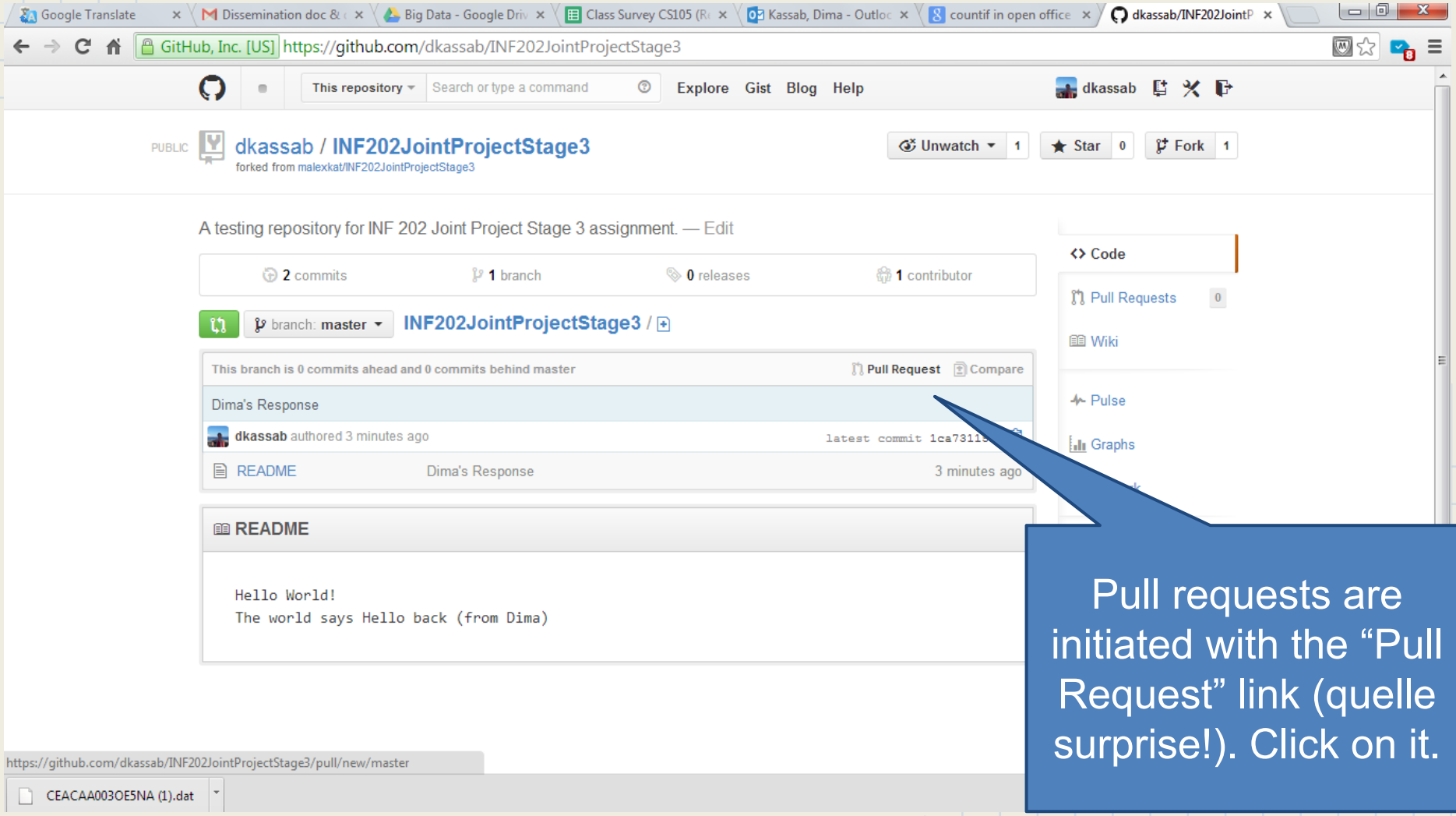
Check Changes on GitHub



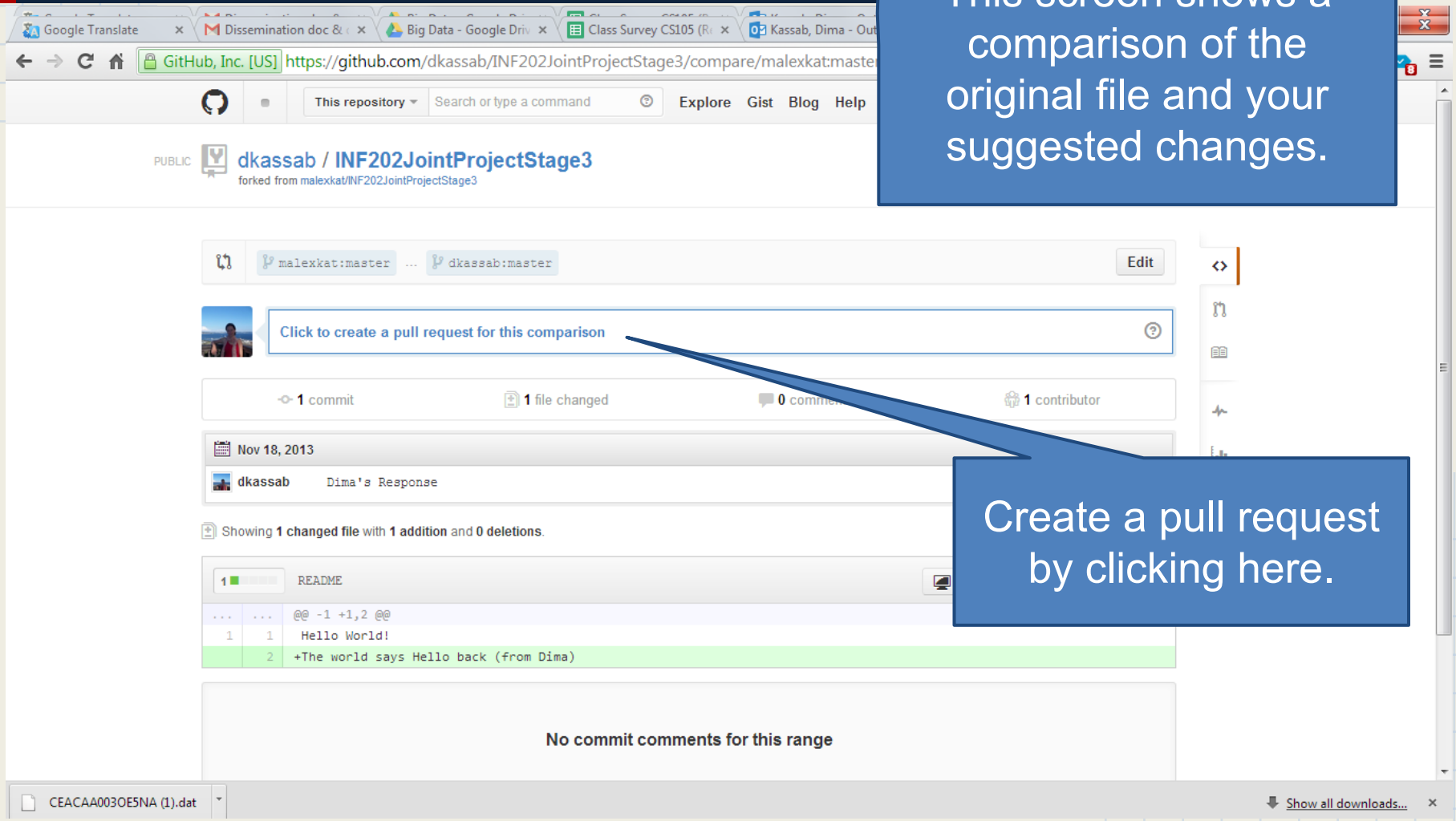
Pulling to Original Repository

- Now that you've modified the README file in your forked repository, it's time to send those modifications to the original repository.
- That's how you contribute to another person's project.
- You ask the repository's owner to review and accept your changes using a "pull" request.

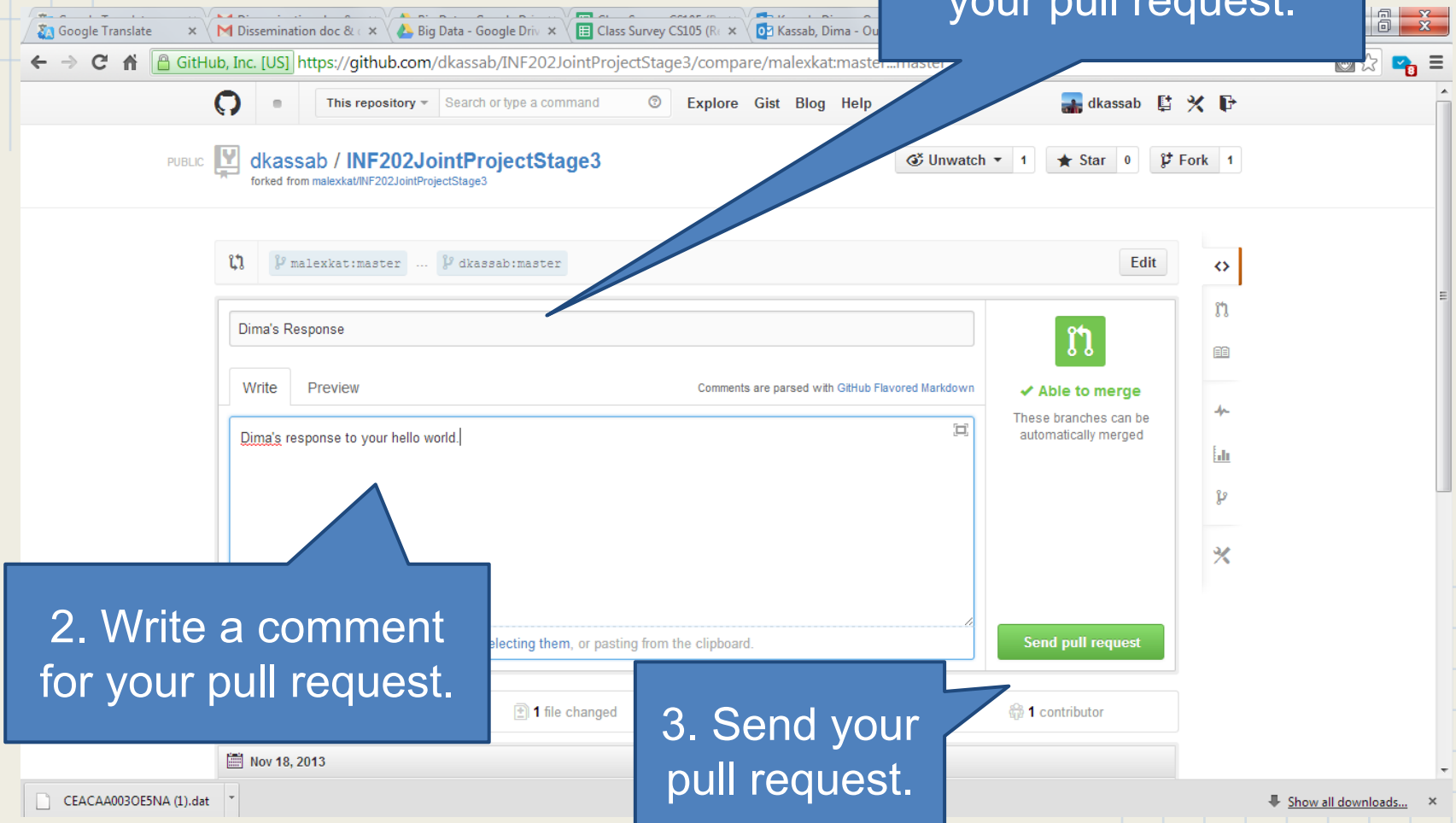
Pull Request



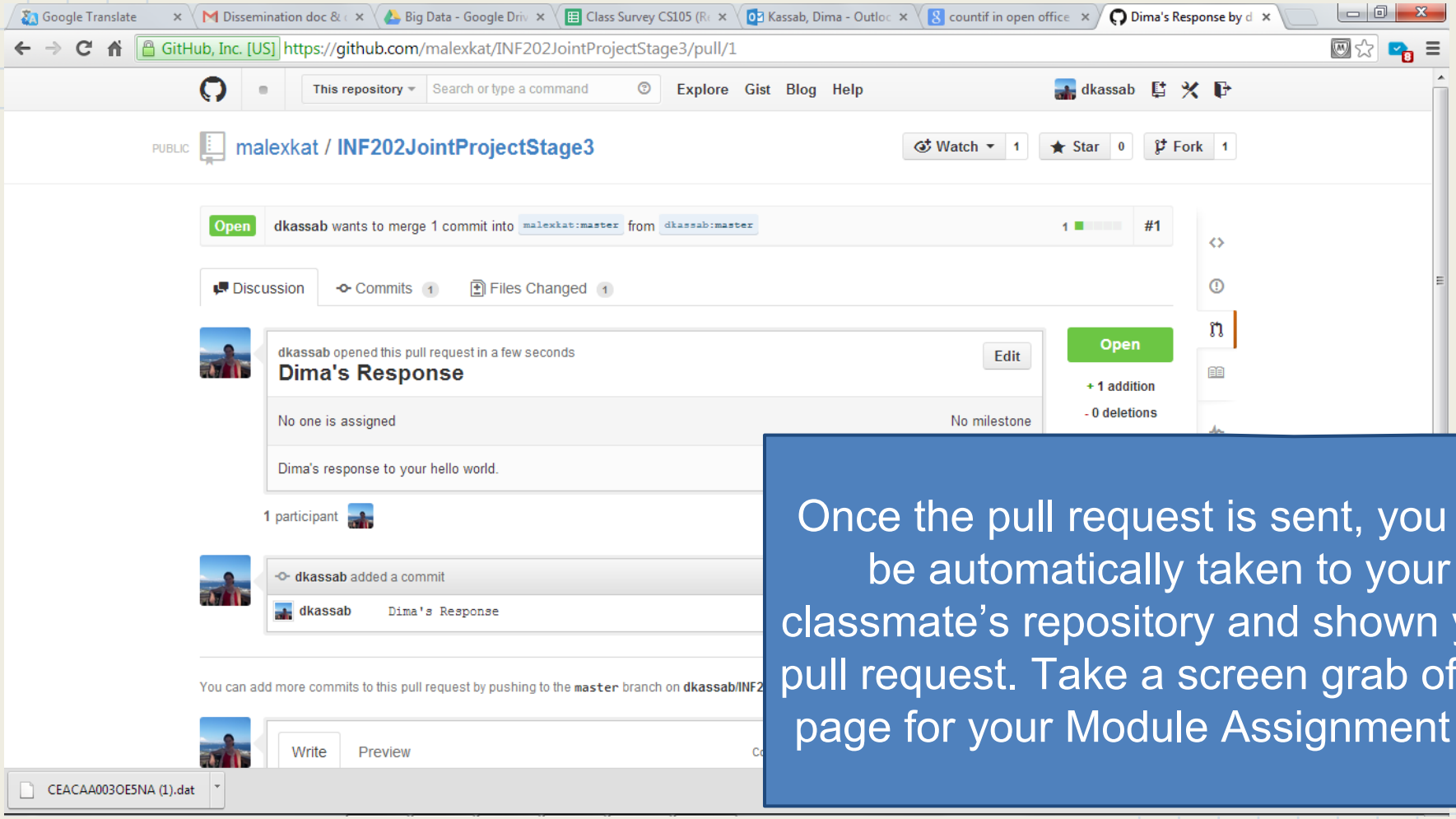
Pull Request



Pull Request



Pull Request Confirmation



Once the pull request is sent, you will be automatically taken to your classmate's repository and shown your pull request. Take a screen grab of this page for your Module Assignment 1d.

Iterate for Learning

- Go back to slide 14.
- Pick a different classmate from the GitHub Assignments spreadsheet. List your name as contributor for that classmate's project.
- Repeat the process of contributing, this time with the new classmate.
- Take screen grabs of your second pull request as detailed in slide 35-36. Make those images part of your Module Assignment 1d submission.

Responding to Pull Requests

- At some point before or after you complete your work above, you should get notice that one of your classmates has submitted a pull request to your repository.
- The second portion of this exercise involves responding to a GitHub pull request.
- In the following slides (39-45), Alex responds to Dima's pull request.

Notice of Pull Requests

[INF202JointProjectStage3] Dima's Response (#1) Inbox x

dkassab <notifications@github.com> Nov 18 (11 days ago)
to malexkat/INF20.

Dima's response to your hello world.

You can merge this Pull Request by running

```
git pull https://github.com/dkassab/INF202JointProjectStage3 master
```

Or view, comment on, or merge it at:

https://github.com/malexkat/INF202JointProjectStage3/pull/1

Commit Summary

- Dima's Response

File Changes

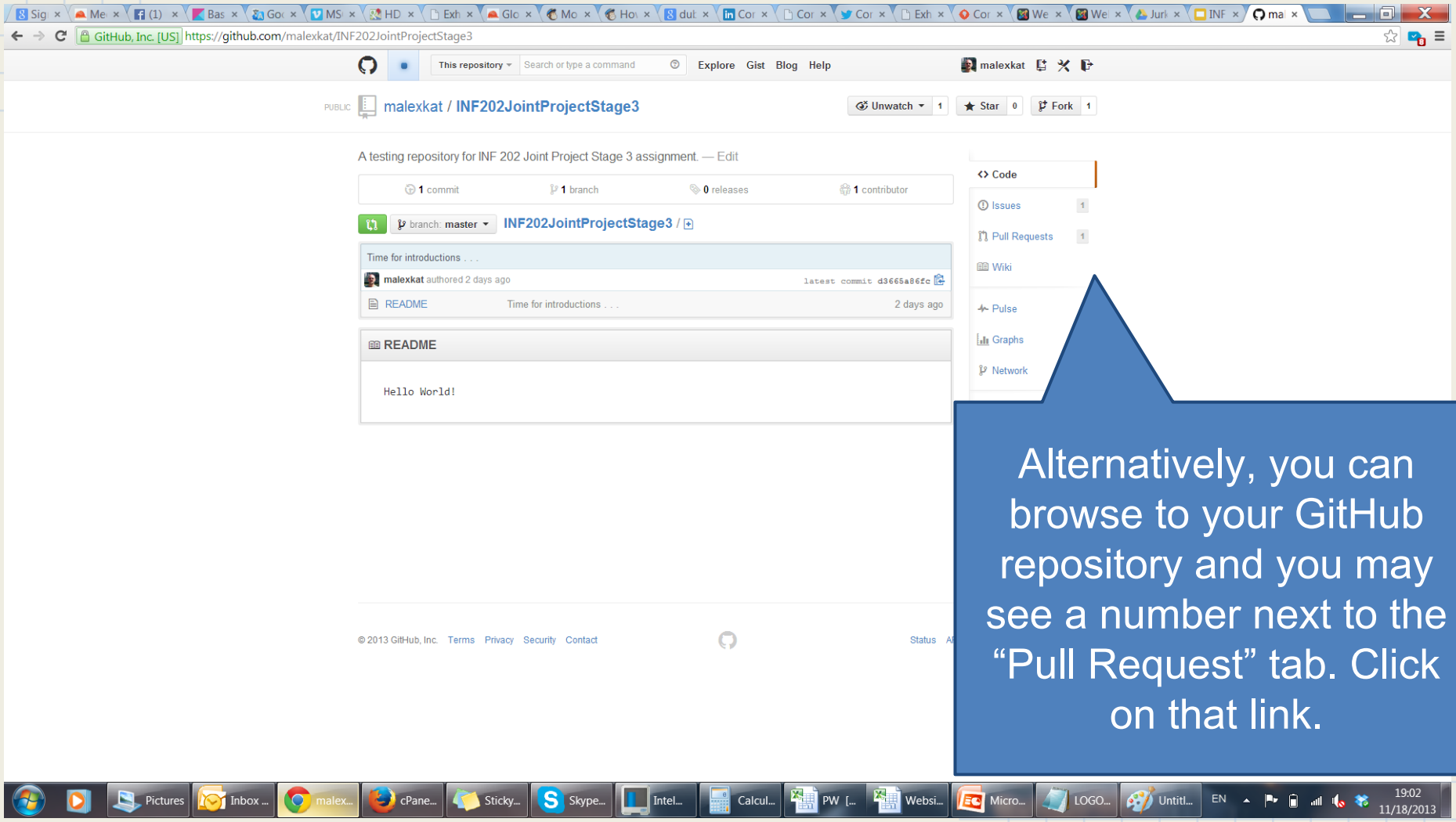
- **M** README (1)

Patch Links:

- https://github.com/malexkat/INF202JointProjectStage3/pull/1.patch
- https://github.com/malexkat/INF202JointProjectStage3/pull/1.diff

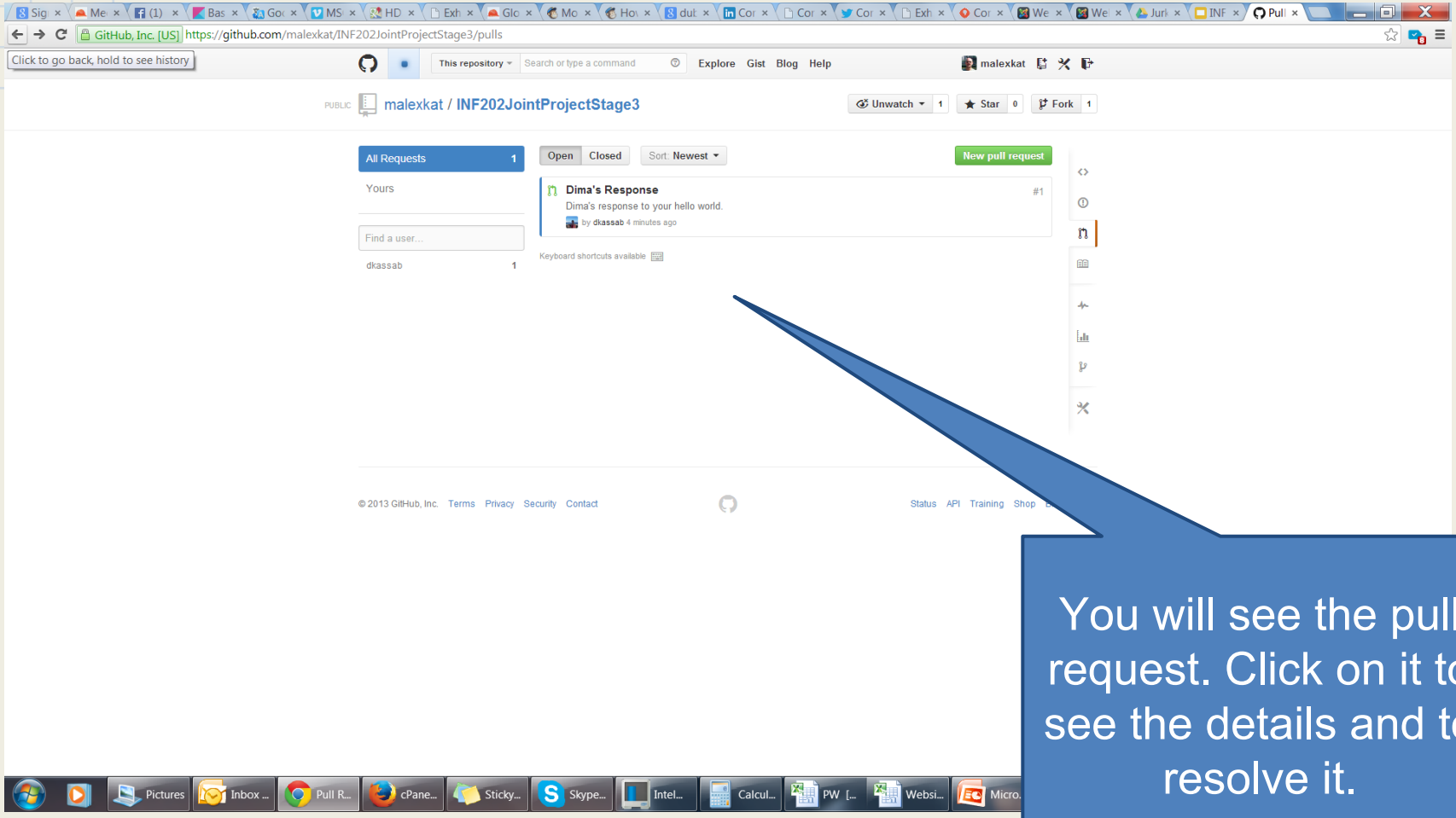
Assuming you cleared communications channels in slides 12-13, you should get an email notice when one of your classmates sends you a pull request. Click on the “view, comment on, or merge it at” link.

Notice of Pull Requests



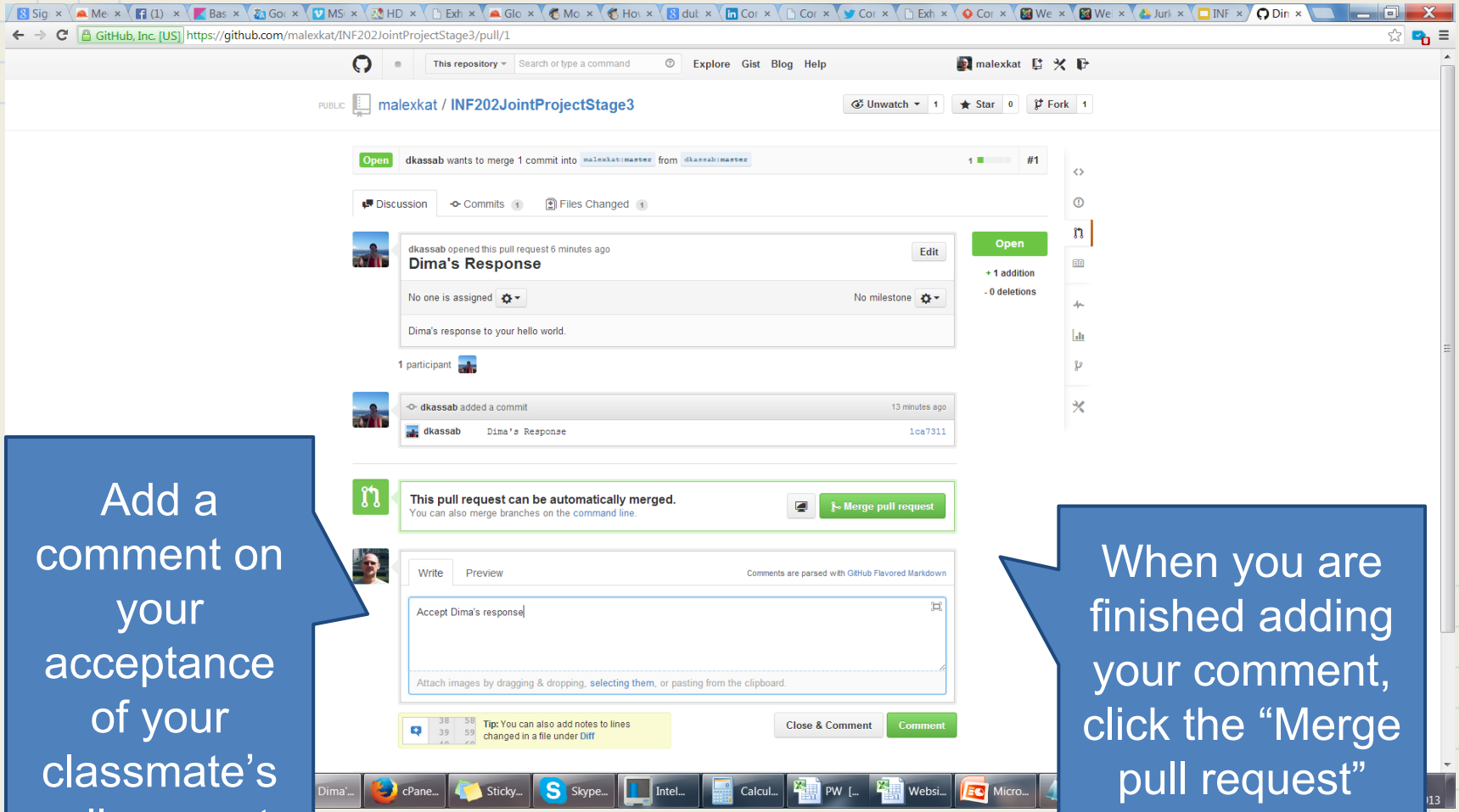
Alternatively, you can browse to your GitHub repository and you may see a number next to the “Pull Request” tab. Click on that link.

Pull Request



You will see the pull request. Click on it to see the details and to resolve it.

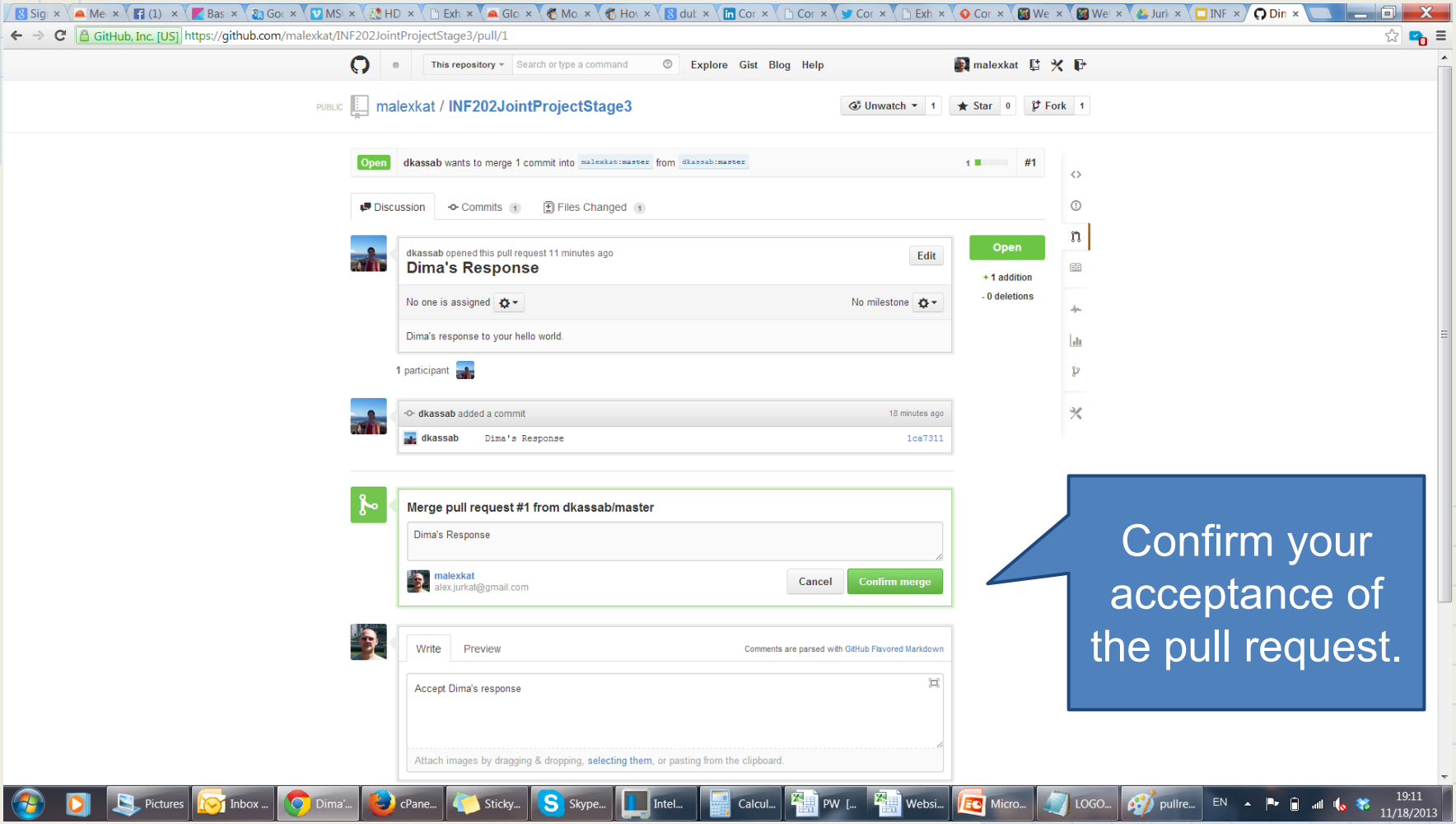
Pull Request Details



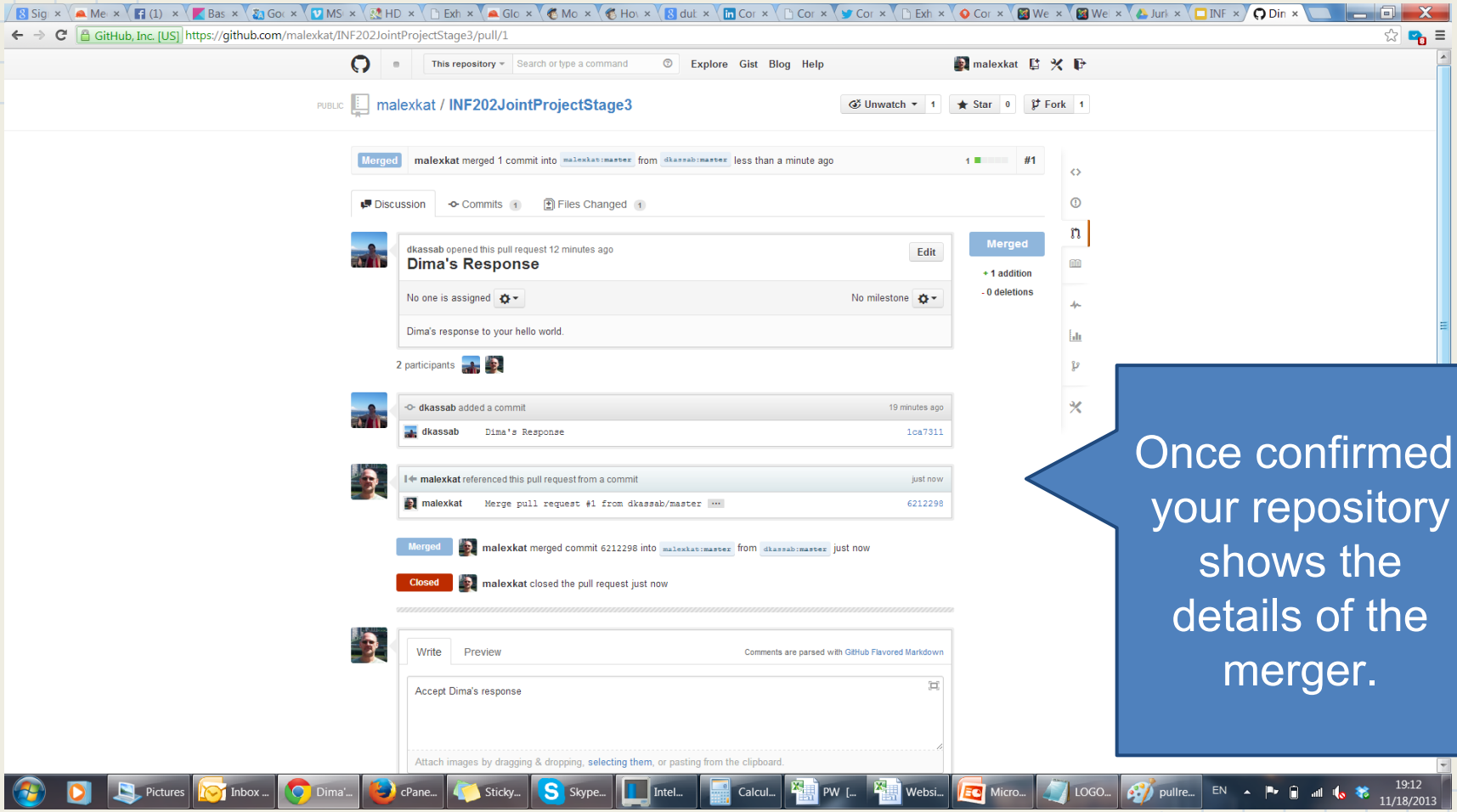
Add a comment on your acceptance of your classmate's pull request.

When you are finished adding your comment, click the "Merge pull request" button.

Confirm Acceptance

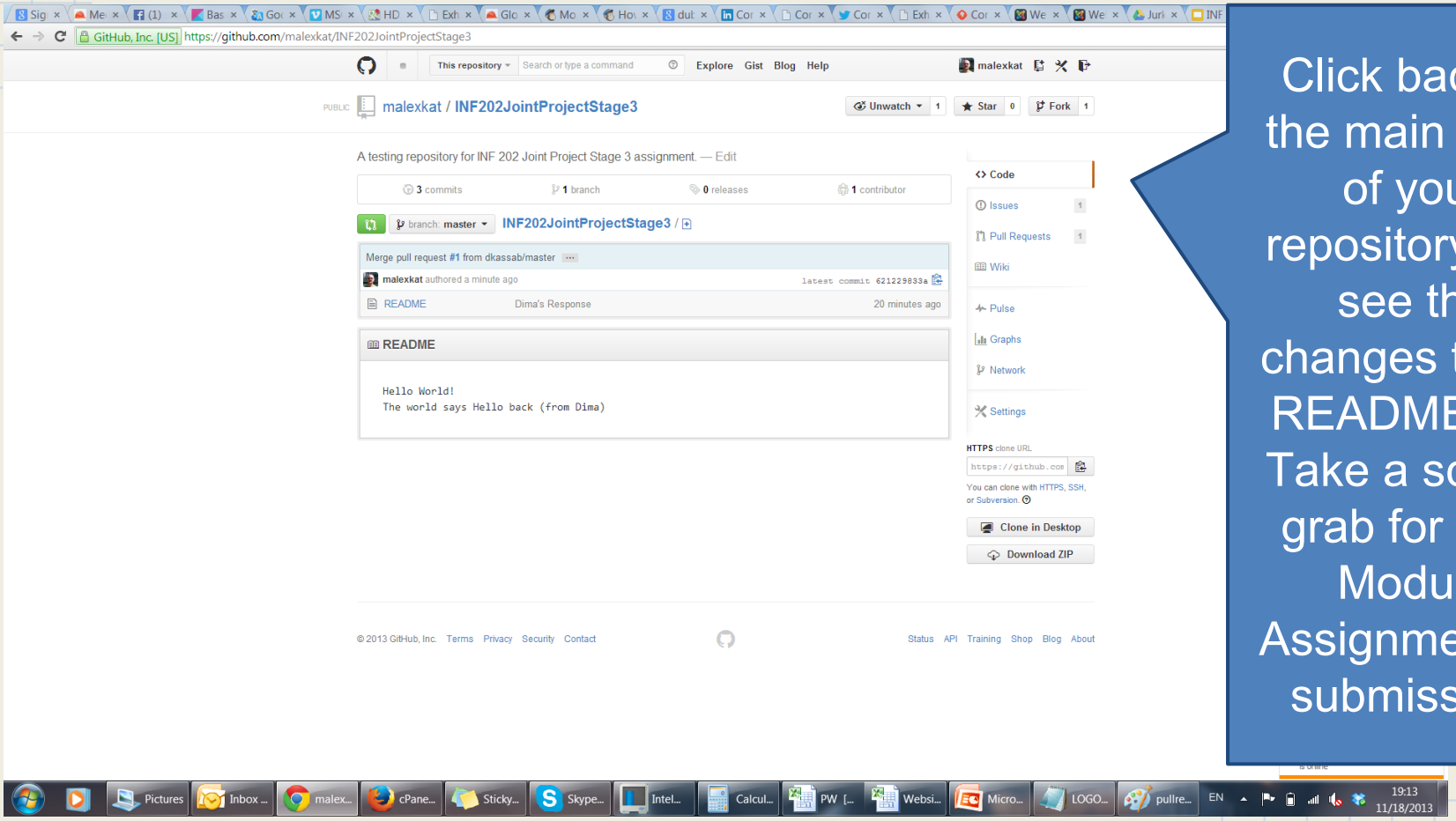


Pull Request Merged



Once confirmed, your repository shows the details of the merger.

Repository Jointly Created!



Click back to the main page of your repository and see the changes to the README file. Take a screen grab for your Module Assignment 1d submission.

Module Assignment 1d

- Combine your screen grabs from slides 35 and 36 (both iterations), plus the screen grab(s) from slide 45 (you may have more than one pull request to resolve).
- Submit these screen grabs as part of your Module Assignment 1d on Blackboard.

Congrats!

You are well on your way to Git and GitHub mastery!