

MAKING MARKING AWESOME!

INTRODUCTION

This session introduces some tools that we can use to:

- Share skeleton code with our students and collect it back in easily.
- Keep track of the changes and progress the students have made.
- Provide feedback on the student's work.
- Automate the testing of whether the students' code does what it should.

We will be using:



GitHub to store, share and comment on the code.



Git to download the code, keep a log of changes to the code, and then upload it again.



Travis CI to automate the testing of any code that students upload.

These tools are all industry-standard and used by the likes of Google, Microsoft, Facebook etc.

GETTING STARTED

- Open up your web browser, navigate to <https://github.com> and sign up.
- Once signed up and in, navigate to <https://github.com/casneconference>

STUDENT GUIDE

COPYING THE CODE ON GITHUB

- Navigate your web browser to <https://github.com/casneconference/example-problem>
- Make your own copy of the code by clicking the fork button on the top right of the page:



DOWNLOADING THE SKELETON CODE

- Open up 'Git bash', which will open up a small terminal screen.
Some useful commands:
 - **ls** lists the contents of the current directory (folder)
 - **cd destination** changes the directory you are in to the destination
 - **mkdir directory-name** makes a new directory
- If you are downloading the code for the first time (like today):
 - **cd** to the directory where you want to download the code to (there is no need to create a subfolder, this will be done automatically).
 - We are going to **clone** the repository from github with the following command:
git clone <https://github.com/username/example-problem>
- If you have already downloaded the code, we can just **cd** to the directory and continue working on it.
- If we have made changes to the code elsewhere and want to update our downloaded code to the latest version, just use the command: **git pull**

You will need git to upload your code again later, so keep it open for now.

COMPLETE THE PROBLEMS

- Open up the file(s) containing the problem set(s) in your text editor and work away as normal.
- The example problem contains tests that we can use to see whether our code does what we want it to. After completing and running your code, calling **run_tests()** will automatically test it.

UPLOAD YOUR CHANGES

- Open git back up.
- If you cloned the repository earlier, you will need to **cd** into it: **cd example-problem**
- **git status** will show a summary of the files that have been added/removed/modified.
- **git add filename** will save the changes you've made to that file ready for upload.
- **git add --all** is a shortcut to add all of your changes in one go.
- **git commit** creates a snapshot of the changes you've added. Make sure to include a message by running it as **git commit -m "A description of the changes"**
- **git push** will then 'push' your changes up to github.

Usually, the workflow is as simple as:

```
git add --all
```

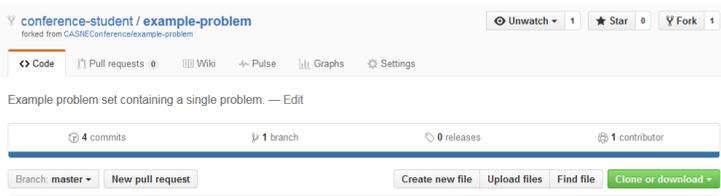
```
git commit -m "A description of the changes"
```

```
git push
```

You will be prompted for your github username and password. Enter these and your changes will be uploaded to github.

SUBMITTING YOUR WORK

- Open up your repository on github (make sure it is your copy, not the original).
- Click the 'New Pull Request' button (bottom left in the picture).



- Add a title, if you are stuck with anything you can use the comments to ask for feedback.
- Submit your work by clicking **Create Pull Request**
- If you need to make further changes to your code, you just need to push a new commit (as above) and it will update automatically.

Finally, make sure your notification settings are set to notify you of any changes. This can be accessed by clicking your profile picture at the top right corner of github, and going to settings.

SUMMARY

- **Fork** the problem set on github if you haven't already done so.
- At the start of a coding session:
 - Open up git bash and **cd** to where your code is stored.
 - **git clone** <https://github.com/username/repositoryname> if you don't have the code locally.
 - **git pull** if you do have the code locally and need to update it with any changes you've made.
- Edit the code to solve the problems. Run it and call **run_tests()** to test it.
- At the end of the session:
 - **git add --all** to add your changes
 - **git commit -m "Description of changes"** to create a new snapshot of your changes.
 - **git push** to upload your changes to github.
- If your work is ready to submit, create a new **Pull Request** on github.

TEACHER GUIDE

SETTING UP A NEW PROBLEM SET

- Create a new repository on github.
- Clone it using git: `git clone https://github.com/username/repositoryname`
- Copy the files from <https://github.com/casneconference/template> to your new repository.
- Edit the readme to provide any notes you want to provide for your students.
- Add in as many problem files as you want. Include doctests at the start so that they can be automatically checked by travis.
- Upload your changes to github:
 - `git add --all`
 - `git commit -m "Problems added"`
 - `git push`
- Log on to travis ci (<https://travis-ci.org/>) with your github username and turn on testing for your new repository.
- Sit back and wait for work to be submitted!