

Git/GitHub Crash Course

CS 420

Eclipse comes with a Git perspective, but I haven't quite figured it out, so here's how to use Git from the command line. Want more info on Git? Use a search engine — there's a lot out there.

1. Global Git configuration, performed once per machine you work from.

```
# Replace my user information with _your_ user information!

git config --global user.name "Tom Kelliher"
git config --global user.email "kelliher@goucher.edu"

git config --global push.default "simple"

# Replace my preferred editor with _your_ preferred editor! This is the
# editor used to create commit messages.

git config --global core.editor "emacs"

# It can be pretty annoying to have to enter your GitHub credentials
# every single time you interact with your remote, so let's get some
# help. Run one of these two commands:

# 1) Cache credentials in memory for one hour (units are seconds). Adjust
# the timeout as you see fit.

git config --global credential.helper "cache --timeout=3600"

# 2) Store credentials _unencrypted_ on disk _permanently_
# Default storage file is ~/.git-credentials

git config --global credential.helper store
```

2. Only one team member runs the following commands, once per new repository. In this example, I'm starting with the HW2d starter files.

```
unzip hw2dStarter.zip
cd Hw2d

# Create a new, empty repository.
git init
```

```
# Stage the current directory and everything underneath it for
# commitment to the local repository.
git add .

# Commit the staged files to the local repository.
git commit -m "Initial commit."

# Associate a remote repository to this local repository.  Origin
# is a handle for the remote's URL.
git remote add origin YOUR_GITHUB_REPO_URL

# Push the local repository to the master branch of the remote
# repository.
git push -u origin master
```

3. Working with a repository.

```
# The other team members run the following command once to clone the
# remote repository.
git clone YOUR_GITHUB_REPO_URL

# If someone else on the team pushes changes to the remote, use this
# to pick them up.
git pull

# If you make changes that you want your team members to have, use the
# following to push changes.
git add A_CHANGED_FILE      # Repeat as necessary.
git commit                  # Commit to local repository.
git push                    # Push commits to remote repository
                             # (origin master).

# To see the current status of your working files, run
git status

# Create the file README.md, add your team members' names to it, along
# with any commentary.

# Add README.md to be staged for committing.
git add README.md

# Use a .gitignore file to tell git to ignore binary files.  Create
# the file, adding the lines
*.o
asst42
# Add other lines as necessary for files that git shouldn't track.
# When finished, add .gitignore to be staged for committing.
```