BUTLER
UNIVERSITY

# CS 142 – Introduction to Computer Science and Programming

## Lecture – 9/4/2019

# Advanced GitHub Topics

# GitHub – Step 1

- Public/Private Key Access
  - HTTPS
    - Requires a handshake each time you access the GitHub repository (username/password).
  - SSH
    - Public/Private key authenticate is done – the public key is stored on the web server and the private key is maintained on your local machine.

https://help.github.com/articles/generating-ssh-keys

# GitHub – Step 2

- Creating A Repository
  - Create a private Git repository on the Butler GitHub.
    - Example: `cs142_fall2019`

- It is easiest to accomplish this through the web front end provided on the Butler GitHub site.

- Make sure to select the **private** option when creating your repository, otherwise everyone can see/access it.

# GitHub – Step 3

- Initial Download of Repository Contents
  - Now that we have our repository setup we need to clone it.
    - Cloning the repository is the primary method for initially downloading the contents of the remote hosted repository onto your local machine in your home directory.
    - We can use Thomas to accomplish this using the following command:

```
git clone
git@github.butler.edu:[username]/cs142
_fall2019.git [destination
_path/directory name]
```

# GitHub – Step 4 (Part I)

- Adding
  - When you clone a Git repository, you have a personal copy of the entire repository on your local machine.
    - An advantage of this approach is that you can **add** and **commit** without having to communicate with the originating server (e.g., Github).
  - Adding a file to your local copy of the repo is simple. We just use the `add` command.
    - `git add [filename]`
  - When you add a file to the local copy, it notifies Git that a file is ready to be committed.

# GitHub – Step 4 (Part II)

- Committing
  - Files are not saved to the repository until you commit it using the commit command.
    - `git commit –m "[Commit Comments Go Here]"`
  - Like with any command line instructions you can add "optional parameters," here we added `–m` which allows for a commit message (comment). The GitHub documentation provides a list of all the parameters available.
  - At this point all of our changes are still local to our repository…we need to push them out to the Butler GitHub.

# GitHub – Step 5 (Part I)

- Pushing
  - We now want to "push" our changes out to the Butler GitHub. To accomplish this we use the Git command of `push`.

```
git push origin master
git push [location] [branch]
```

  - The command above is pushing to the `origin` location of the `master` branch. If you want to push to a different location you would just need to specify the respective parameters.
    - Origin is simply the remote repository.
  - When working with collaborators, they will also push their changes as well. We will eventually want to integrate their changes into our workspace. This is when we want to **pull** from the repository.

# GitHub – Step 5 (Part II)

- Pulling
  - The `pull` command is similar to the `push` command in terms of its syntax. If you do not specify a branch, then it pulls everything that is new and merges the changes into your local repository.

```
git pull origin master
git pull [location] [branch]
```

  - Again, we can specify the specific location and branch to “pull” from.