



Introduction

Self-introduction


What's the course about?

Git and github



What is this Course About?

- To review the fundamental data structures and programming techniques learned in CSCI241 & 240
- To learn more advanced data structures and programming techniques and program development methodologies
- To strengthen your coding skill
- To evaluate, analyze and select data structures and algorithms for solving problems **efficiently**
- To learn to use git/github (source management).

- 
- The CORE course of the entire computer science education!



Course Information

- Course website in Blackboard System
 - Course material
 - Assignment (invitation links from github)
 - Up-to-date grades
- All programming assignments are graded on `turing.cs.niu.edu` and `hopper.cs.niu.edu` (Linux) in `g++`.
- Assignment electronic submission and code management are done by `git/github`.
- Dailies: Grade-O-Matic (more later)
- Recitations



Syllabus walk-through ...



To do list:

Entrance quiz ...

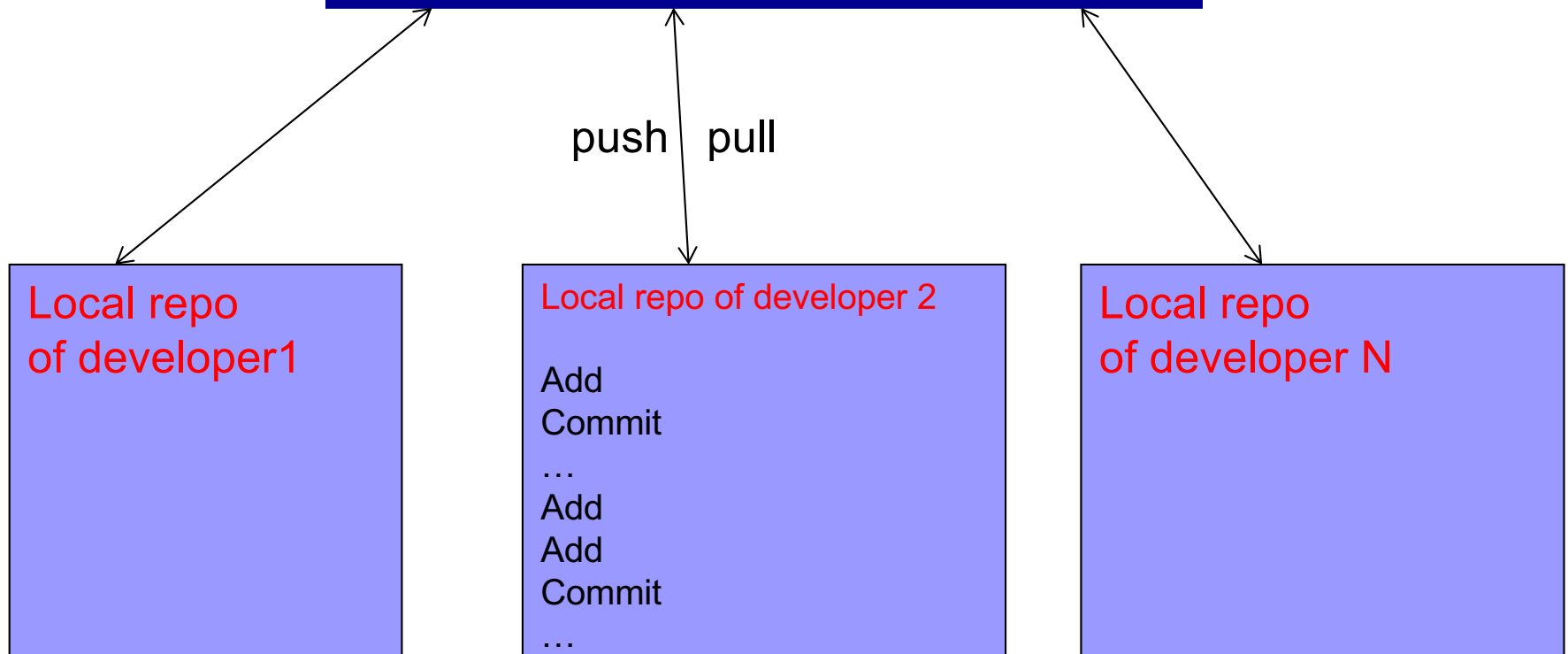
- Make sure you are able to login onto turing/hopper with your z number.
- Make sure you have a github login id, and fill the registration form for the github id, before 09/04.



Lecture 2: Git and Github:

- Background
 - Source Control Management:
 - Version control!
 - Collaboration, and Distributed collaboration!
 - Linus Torvalds did not like some stuff before (such as CVS)
 - Allow many people to collaborate together on a project and keep track of the changes made by everyone.
- Key concepts
 - git (~2005) versus github (~2008)
 - Not the same thing!
 - Github is built on top of git with a website and some UI features
 - Repository (repo in short): tracks file changes to build a history (in .git folder)
 - Local versus remote repository
 - Remote is hosted on the Internet or another network.
 - Branch: a pointer to a snapshot of repo at a timepoint.

REMOTE REPO





For our assignment: REMOTE REPO on github

push

Pull if you have other local repos

Your local repo on turing/hopper

Add
Commit

...

Add
Add
Commit

...



Key git Commands

- **git-clone (git clone):** clone a repo into a new directory
git clone <URL from GitHub repository>
- **git-add (git add):** add file contents
git add assign1.cc
- **git-commit (git commit):** record changes to the repo
- **git-push (git push):** update remote repo
- **git-status (git status):** show the working tree status
- **git-pull (git pull)** Fetch from and integrate with another repository or a local branch (typically not needed for our assignment)
 - *Not to be confused with “pull request”, which is a process for a developer to notify that they have complete a feature, and team members can review the code.*



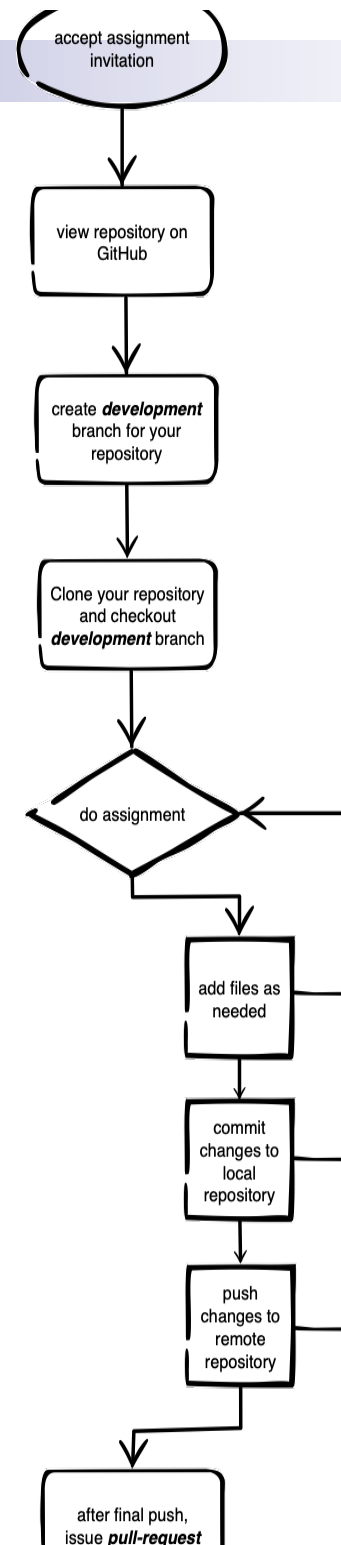
Assignment Submission Flow -- the flow of assignment 1.

1. Accept assignment invitation
2. View repository on github; create a ***development*** branch

important: Do not mis-spell “development” or change cases!!!

3. clone and checkout development branch on your local machine (hopper/turing; need to setup ssh key first.)
4. Do assignment locally: add; commit; push ... loop
5. After final push, issue pull-request on github


A handout with a clearer image for the flow is posted in Blackboard:





Example flow:

- `git clone` <URL from GitHub repository>
- `cd` the directory
- `git checkout development` (or use `switch`)
- Edit some files
- `git add` <file(s) you have created/modified>
- `git commit -m "what you did"`
- Loop editing/add/commit
- `git push`



Authentication (so that you can work with a “remote” repository)

- You can use the `ssh-keygen` program on Turing or Hopper to generate your keys, and copy the `public key` information in your GitHub account *settings*.

- For details:

<https://docs.github.com/en/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent>

- A video for SSH-keygen n 340 by another instructor:

<https://www.youtube.com/watch?v=CdZa8MzLtjQ>

- ☐ Note: support for password authentication was removed on August 13 2021.



Resources

- 2-page git command cheatsheet:
 - <https://education.github.com/git-cheat-sheet-education.pdf>
- One tutorial on youTube:
 - <https://www.youtube.com/watch?v=DVRQoVRzMIY&t=2187s>
- History of git (podcast)
 - <https://www.weave.works/blog/15-years-of-git>

