

O Object O Oriented P Programming

S1 = Python with Bianca

S2 = Java with Bryan

DT228(TU856)/DT282(TU858) - 2



**COMPUTER
SCIENCE**

Lecture Week 1

Introduction Part 2

Learning Objectives

- Answer student questions (I've gotten some in email)
- Review your knowledge of C
- Get to know version control using GitHub

Student Questions

- Where is the material?
- What is the pass rate?
- Can I swap lab groups?
 - Tutors?

Now to you

- Test your knowledge of year 1 programming as a foundation for this year



Poll Time

GitHub

GitHub, Inc. [US] | github.com/gitpitch/gitpitch

Code Issues 15 Pull requests 1 Projects 0 Wiki Security Insights

Join GitHub today Dismiss

GitHub is home to over 40 million developers working together to host and review code, manage projects, and build software together.


[Sign up](#)

The Markdown Presentation Service For Developers on Git <https://gitpitch.com>

[markdown](#) [presentations](#) [slideshows](#) [git](#)

216 commits 3 branches 3 releases 8 contributors MIT

Branch: master New pull request Find File Clone or download

 gitpitch Slide deck doc links updated. Latest commit 660a859 on 6 Aug

app	Added Video delim loop + muted params.	10 months ago
assets	README updates.	2 months ago
conf	Highlight.js upgrade to v9.13.1.	10 months ago
deps	Updated decktape intergration.	2 years ago
images	Integrated tweaks for PITCHME.md copy and visual styles (#118)	2 years ago
project	Disable unused sbt plugins	2 years ago
public	Highlight.js upgrade to v9.13.1.	10 months ago

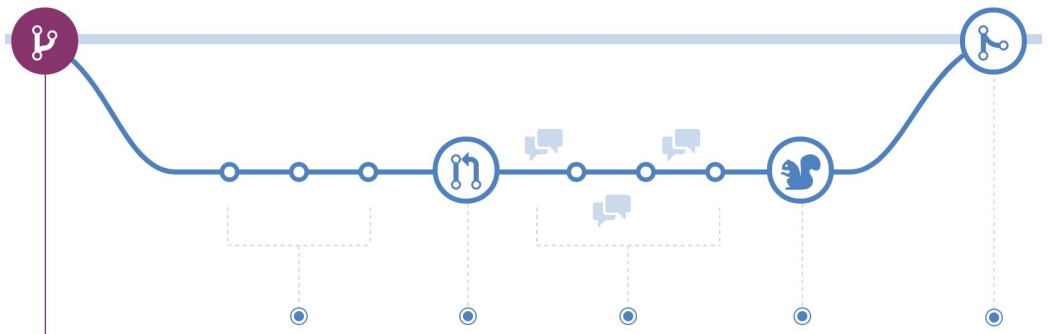
Why version control is important

- Manages different versions of your code
 - Multiple versions of code
 - Multiple collaborators
 - Manages changes
- Changes are traceable and reversible
- Clarity
- Efficiency
- Reduced duplication
- Reduced errors

Git/GitHub

- git is a command line tool
- GitHub is a webfrontend that works on top of git

- Used for version control of code
- Enables teams to work together on the same piece of code in a controlled manner
 - How would this look like when using files?
- Need git on your computer, install [here](#)
- Need a [GitHub account](#) (*we will do this together in the lab*)
- You can control it just via GUI and there are many third party tools available and many IDEs now directly link in with GitHub
 - To understand how it works it's better to start off with the command line, later you can decide how you want to use it



Getting started (we do this in the lab)

1. Install git

- a. Mac: <http://git-scm.com/download/mac>
- b. Windows: <http://git-scm.com/download/win>
- c. Linux (debian): `> sudo yum install git`

2. Create a github account www.github.com

3. Get working, see later slide

- Linus Torvald created git to aid with Linux kernel development
- There are many different version control systems
- Github is a web hosting repository for git, there are others, such as bitbucket
 - Hosts remote repos
 - Enables collaboration
 - Extra features on top of git

Important Terminology (1/5)

- Branch is the main concept
- Anything in the **master** branch is always deployable
- **New** branches should be created off of the **master** branch

- **Repository**: A repository is the most basic element of GitHub. They're easiest to imagine as a project's folder. A repository contains all of the project files (including documentation), and stores each file's revision history. Repositories can have multiple collaborators and can be either public or private.
- **Branch**: A branch is a parallel version of a repository. It is contained within the repository, but does not affect the primary or master branch allowing you to work freely without disrupting the "live" version. When you've made the changes you want to make, you can merge your branch back into the master branch to publish your changes.
- **Upstream**: When talking about a branch or a fork, the primary branch on the original repository is often referred to as the "upstream", since that is the main place that other changes will come in from. The branch/fork you are working on is then called the "downstream".

- Try out new ideas in a separate branch.
- Choose descriptive names!!

[3]

Terminology cont'd (2/5)

- **Clone**: *A clone is a copy of a repository that lives on your computer instead of on a website's server somewhere, or the act of making that copy. With your clone you can edit the files in your preferred editor and use Git to keep track of your changes without having to be online. It is, however, connected to the remote version so that changes can be synced between the two. You can push your local changes to the remote to keep them synced when you're online.*
- **Commit**: *A commit, or "revision", is an individual change to a file (or set of files). It's like when you save a file, except with Git, every time you save it creates a unique ID (a.k.a. the "SHA" or "hash") that allows you to keep record of what changes were made when and by who. Commits usually contain a commit message which is a brief description of what changes were made.*

- Each commit is a separate unit of change
- Shows history of your programme
- **Write clear commit messages!**

[3]

Terminology con'd (3/5)

- **Diff**: A diff is the difference in changes between two commits, or saved changes. The diff will visually describe what was added or removed from a file since its last commit.
- **Fetch**: Fetching refers to getting the latest changes from an online repository without merging them in. Once these changes are fetched you can compare them to your local branches (the code residing on your local machine).
- **Fork**: A fork is a personal copy of another user's repository that lives on your account. Forks allow you to freely make changes to a project without affecting the original. Forks remain attached to the original, allowing you to submit a pull request to the original's author to update with your changes. You can also keep your fork up to date by pulling in updates from the original.

[3]

Terminology cont'd (4/5)

- **Merge**: *Merging takes the changes from one branch (in the same repository or from a fork), and applies them into another. This often happens as a pull request (which can be thought of as a request to merge), or via the command line. A merge can be done automatically via a pull request via the GitHub web interface if there are no conflicting changes, or can always be done via the command line.*
- **Pull**: *Pull refers to when you are fetching in changes and merging them. For instance, if someone has edited the remote file you're both working on, you'll want to pull in those changes to your local copy so that it's up to date.*
- **Pull request**: *Pull requests are proposed changes to a repository submitted by a user and accepted or rejected by a repository's collaborators. Like issues, pull requests each have their own discussion forum.*

[3]

- Can be opened at any stage of the development process.
- Everyone can look at suggested changes
- **Pull requests can be used to ask for feedback**

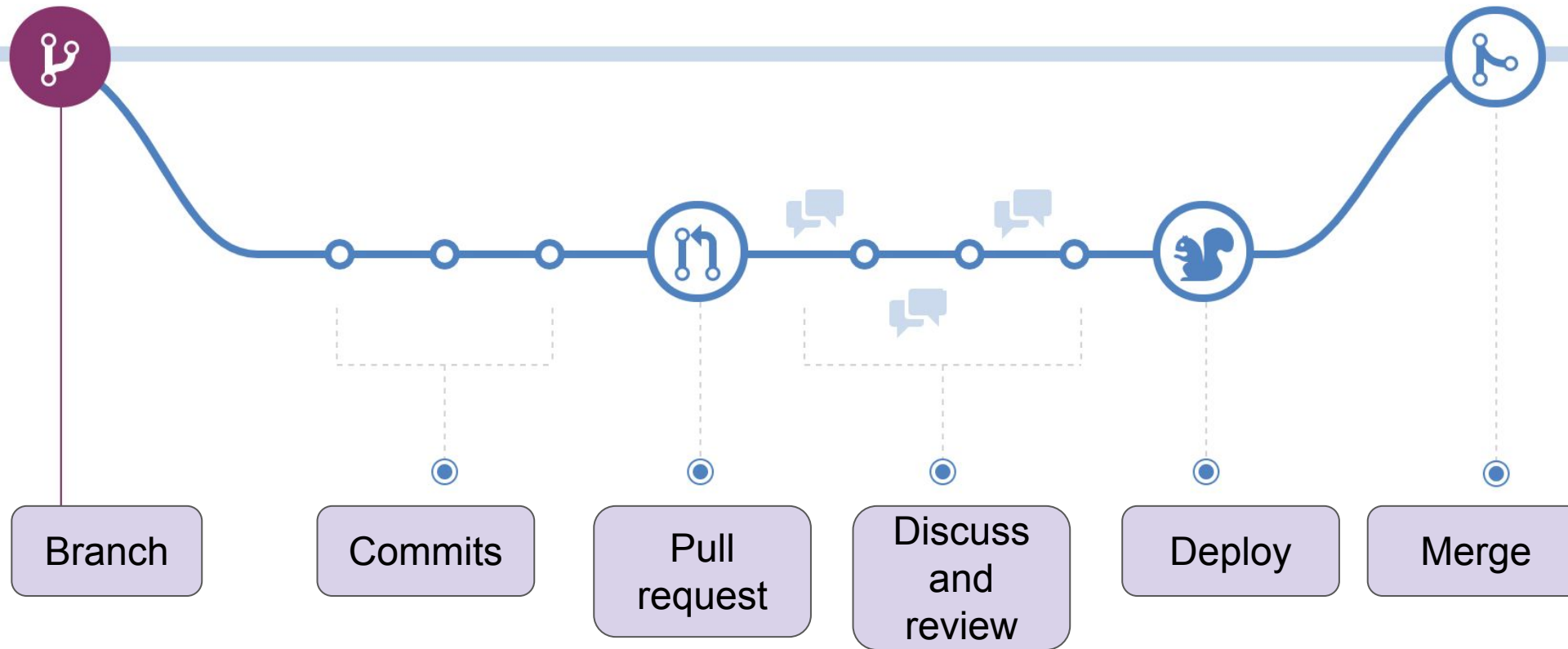
Terminology cont'd (5/5)

- **Push**: *Pushing refers to sending your committed changes to a remote repository, such as a repository hosted on GitHub. For instance, if you change something locally, you'd want to then push those changes so that others may access them.*
-

This is not an extensive list,
explore for yourself!

[3]

GitHub process



Master Branch Name Change

- Change from master to main
- Default from 1. Oct 2020
 - No change until then

[8]

The Steps of git

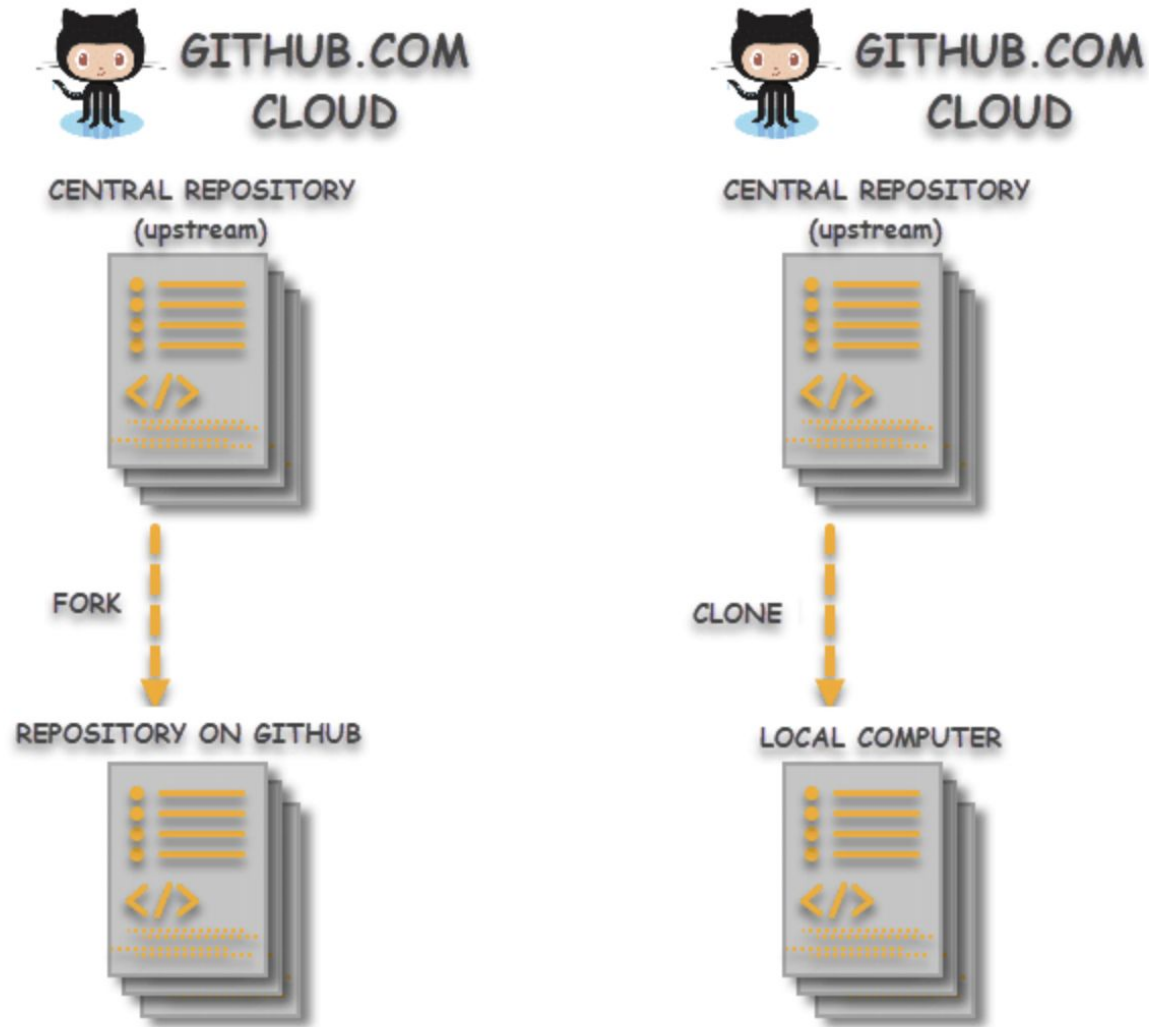
1. Create a new repository
2. Add a new file to the repo
3. Add a file to the staging environment
4. Create a commit
5. Create a new branch
6. Create a new repository on GitHub
7. Push a branch to GitHub
8. Create a pull request (PR)
9. Merge a PR
10. Synch changes on the GitHub repo with your local computer

You can use GitHub and other git online platforms without needing to know about the command line. However, many employers will expect you to know command line tools as well.

Example usage will be demo-ed on Wednesday in the tutorial!

Poll time

Solution



[6, 7]

Summary

- ★ **Revision C**
- ★ **Background Python**



References

1. Moutaz Haddara, Introduction to Object-oriented programming, slideshare, 2014.
2. Jamie Chan, Learn Python in one day, 2014.
3. The github glossary, <https://help.github.com/en/articles/github-glossary>, accessed Sep 2020.
4. Introduction to GitHub, Rakesh Sukumar, 13 Sep 2015, accessed via slideshare Sep 2019.
5. Github guides, <https://guides.github.com/introduction/flow/>, accessed Sep 2020.
6. Difference Fork and Clone.
<https://www.toolsqa.com/git/difference-between-git-clone-and-git-fork/#:~:text=When%20you%20fork%20a%20repository,with%20the%20help%20of%20Git.,>
accessed Sep 2020.
7. Difference fork and clone, GitHub Community,
<https://github.community/t/the-difference-between-forking-and-cloning-a-repository/10189>, accessed Sep 2020.
8. GitHub Name change,
<https://github.com/github/renaming#:~:text=GitHub%20is%20gradually%20renaming%20the,and%20all%20of%20their%20contributors.>, Accessed Sep 2020