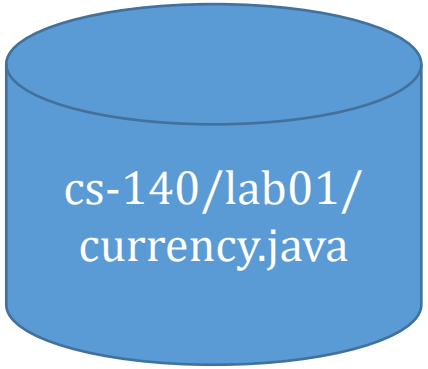
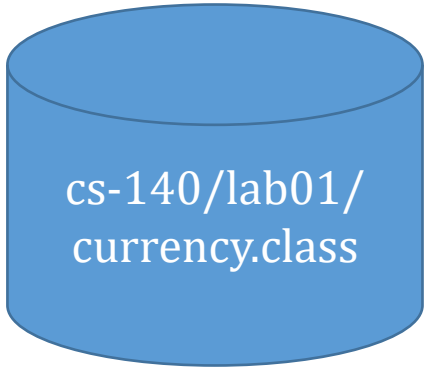
CS – 140
Introduction to Git

Motivation

- You are given an assignment, hw01, which needs three object oriented classes; a bank, an account, and currency.
- You get done with lab early, so you start working on the homework, and create:

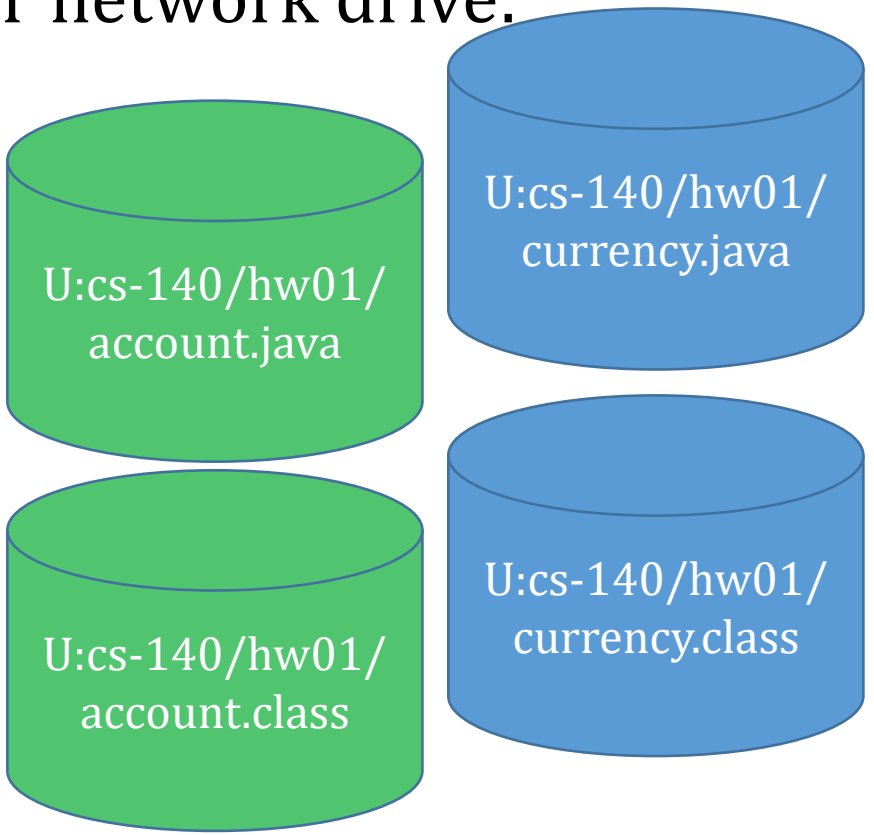




- And compile to create:
- But you want to work more, so you email currency.java to yourself

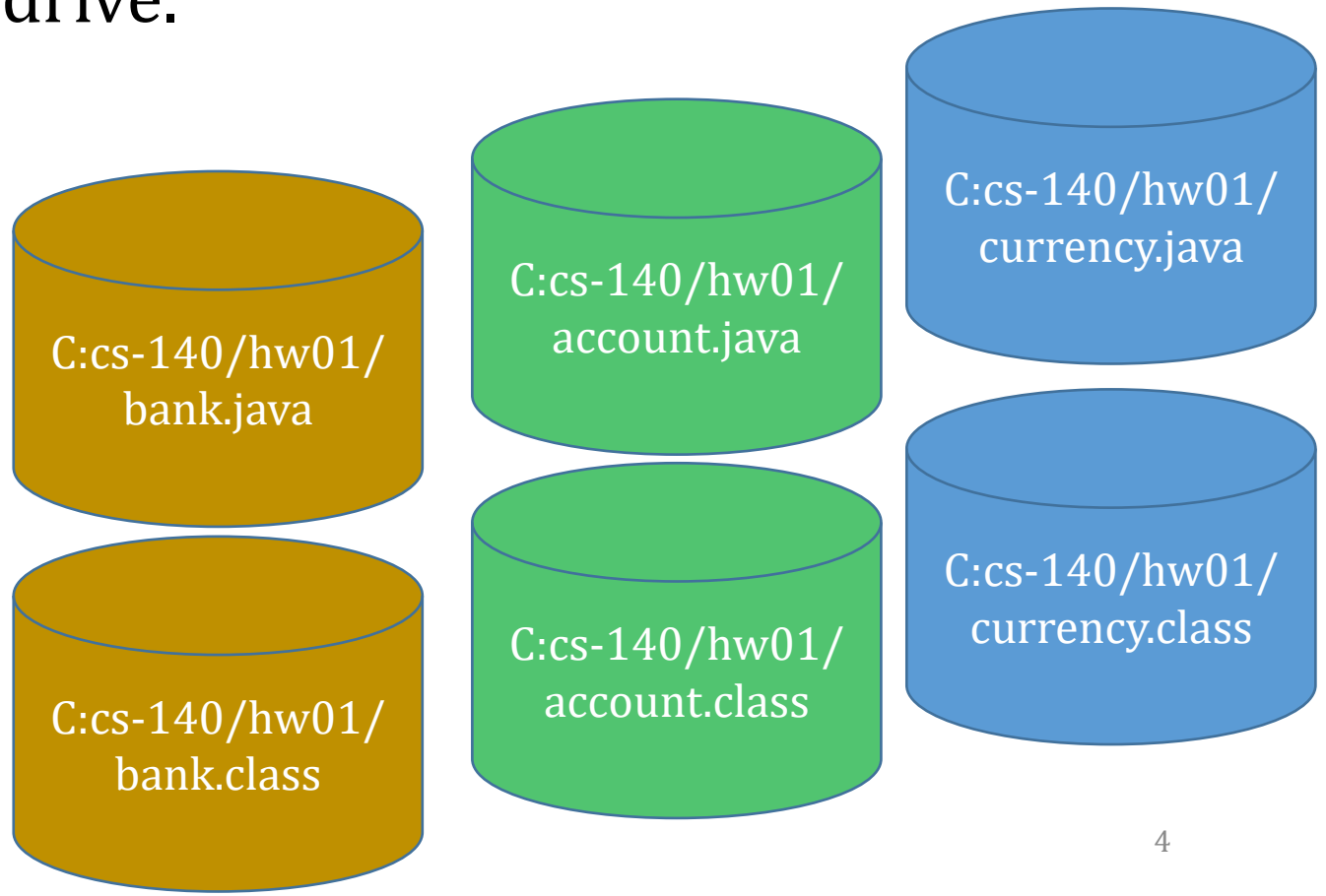
Motivation – 2

- You have some extra time in the library, so you go to a PODS machine, and download currency.java to your network drive:
- And add code for the account:
- And compile to create:
- But you find mistakes in currency.java, so you need to update that file…
  - and recompile



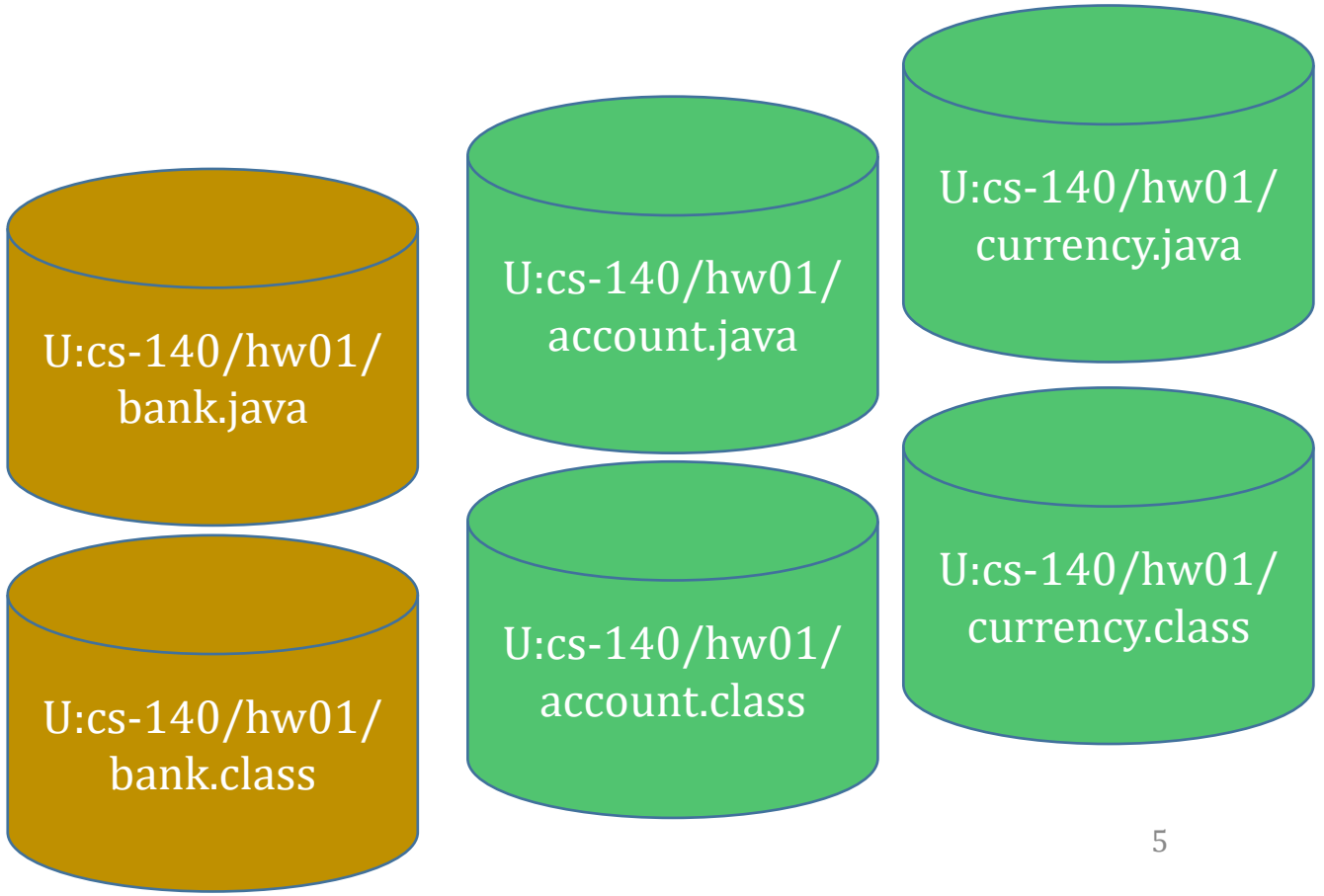
Motivation – 3

- Back in your apartment, you download currency.java, and FTP account.java to your laptop C drive.
- And add code for the bank:
- And compile to create:
- But you find mistakes in account.java, so you need to update that file…
  - and recompile



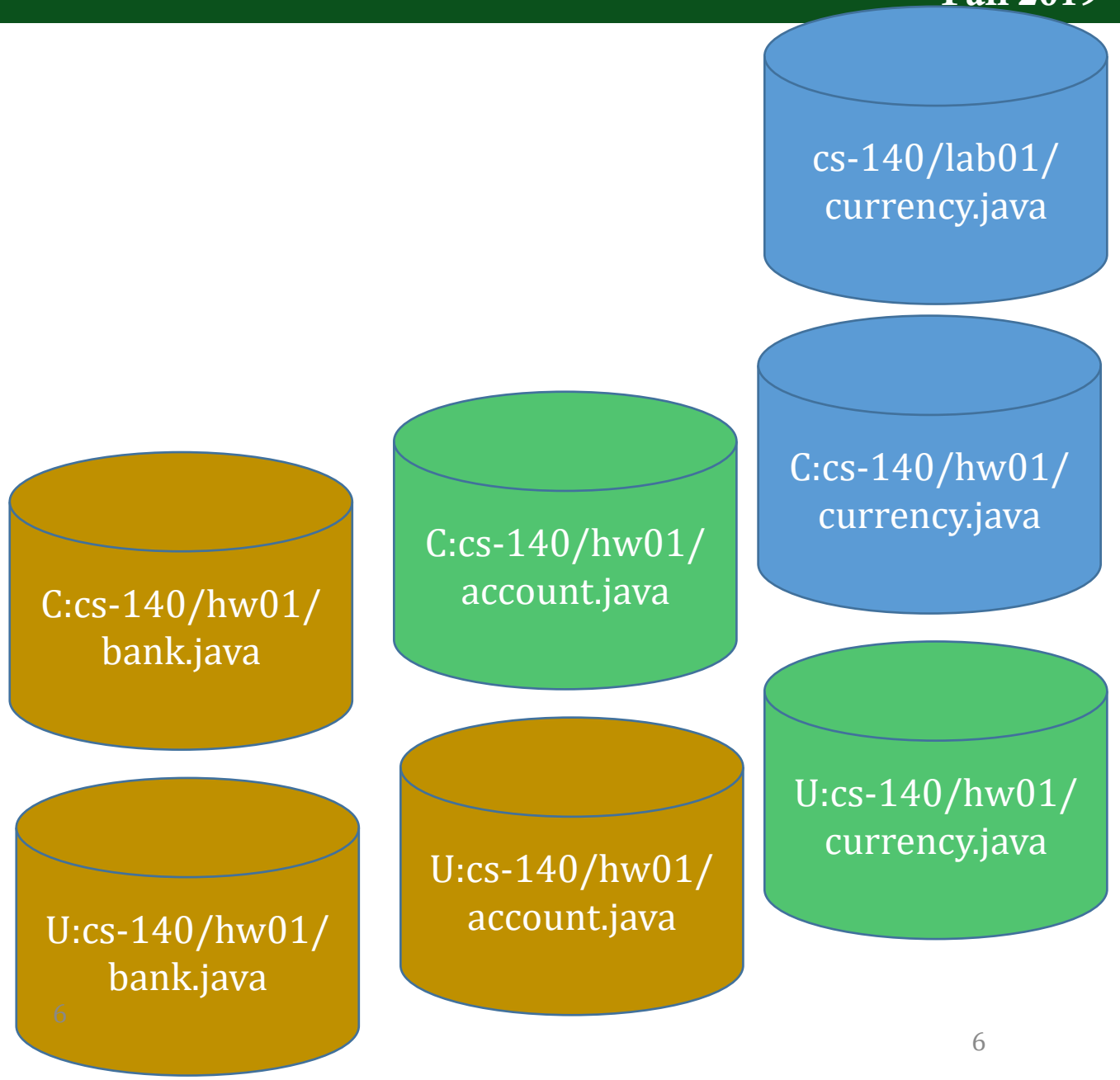
Motivation – 4

- The next day in the library, FTP bank.java to your network U: drive…
- And compile:
- Oops… forgot the changes to account.java… FTP that
- Still doesn't work… oops forgot to recompile!



Motivation – 5

- Now… What do I turn in???



Making Linux

- Linus Torvalds learned about UNIX in 1990 as a graduate student taking an operating systems course
- In 1991, Torvalds went to a Richard Stallman lecture about GNU and free software
- By 1994, Torvalds had released Linux Version 1.0
- Currently, Linux has almost 5,000 programs!
- Manage That!

GIT – Version Control

The name "git" was given by Linus Torvalds when he wrote the very first version. He described the tool as "the stupid content tracker“ and the name as (depending on your way of thinking):

- random three-letter combination that is pronounceable, and not actually used by any common UNIX command. The fact that it is a mispronunciation of "get" may or may not be relevant.
- stupid. contemptible and despicable. simple. Take your pick from the dictionary of slang.
- "global information tracker": you're in a good mood, and it actually works for you. Angels sing, and a light suddenly fills the room.
- "goddamn idiotic truckload of sh*t": when it breaks

Git Disclaimer

- Git was designed to manage open source development of Linux
  - Very complicated scenarios… One developer tweaks some components of Linux… the community needs to decide whether to accept those changes.
  - Are they good changes? Do they interact with other stuff? Are they downward compatible? Does a different developer do the same thing better? etc. etc.
- We only need a tiny portion of Git capabilities for this class!
  - Single or Small group (team) development
  - Single threaded development – One change at a time.

Basic Git Terminology

- Repository : Set of files required to make a single program
  - or set of inter-related programs.
- Master Repository : The reference copy of a repository.
  - Typically not directly editable
- Local Repository : A local copy of the Master Repository
  - Typically where code gets edited
  - A "clone" create a new local repository by copying the master
- commit : make a new version of the code
  - Typically with the changes made to a local repository
- push : Copy committed changes from local to master repository
- pull : Copy committed changes from master to a local repository

Git Workflow

- git clone – make a local copy of the master repository
- edit program files (.java) in the local repository
- test the changes until you are happy with those changes
- git add *filename* – add new files to local repository for git to track
- git commit –a "*comment*" – make a new version locally
- git push – copy your committed changes to the master repository
- git pull – [From a different local repository] copy changes in the master repository to the local repository

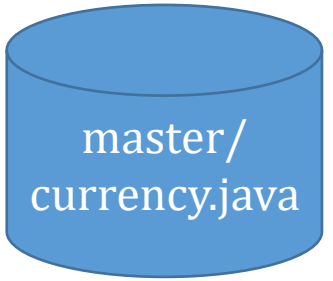
GitHub Service

- Software owned by Microsoft
- Uses “GIT” commands as an interface
- Requires userid/password
- Master repository kept in the "cloud"
  - Has name, owner, permissions, versions, branches
  - Has a URL and web browser interface to look at the master repository
- Public v.s. Private repositories

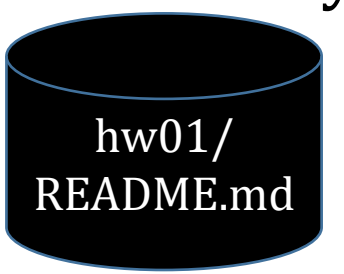
GitHub Classroom

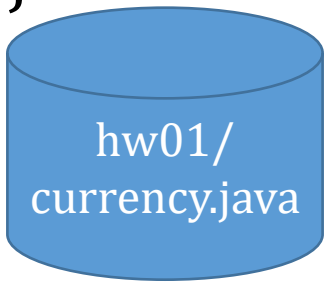
- Software package built on GitHub
- Allows Prof. to create an “assignment”
  - Consists of a repository with a README.md and potentially starter files
  - Has an “invitation” URL
- If you open the invitation, GitHub Classroom will allow you to “Accept” your invitation
  - Create a new private repository for you *assignment-gitId*
  - You may modify your repository to “do” the assignment
  - TA’s and I can see your (modified) repository to grade it

Motivation Scenario with Git

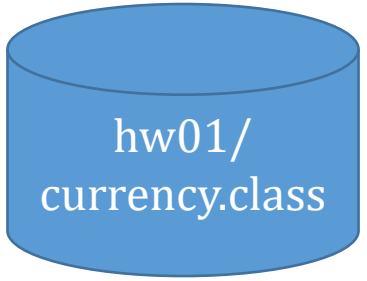


- You are given an assignment, hw01, which needs three object oriented classes; a bank, an account, and currency.
- You get done with lab early, so you start working on the homework: git clone, create currency.java
- And compile to create:

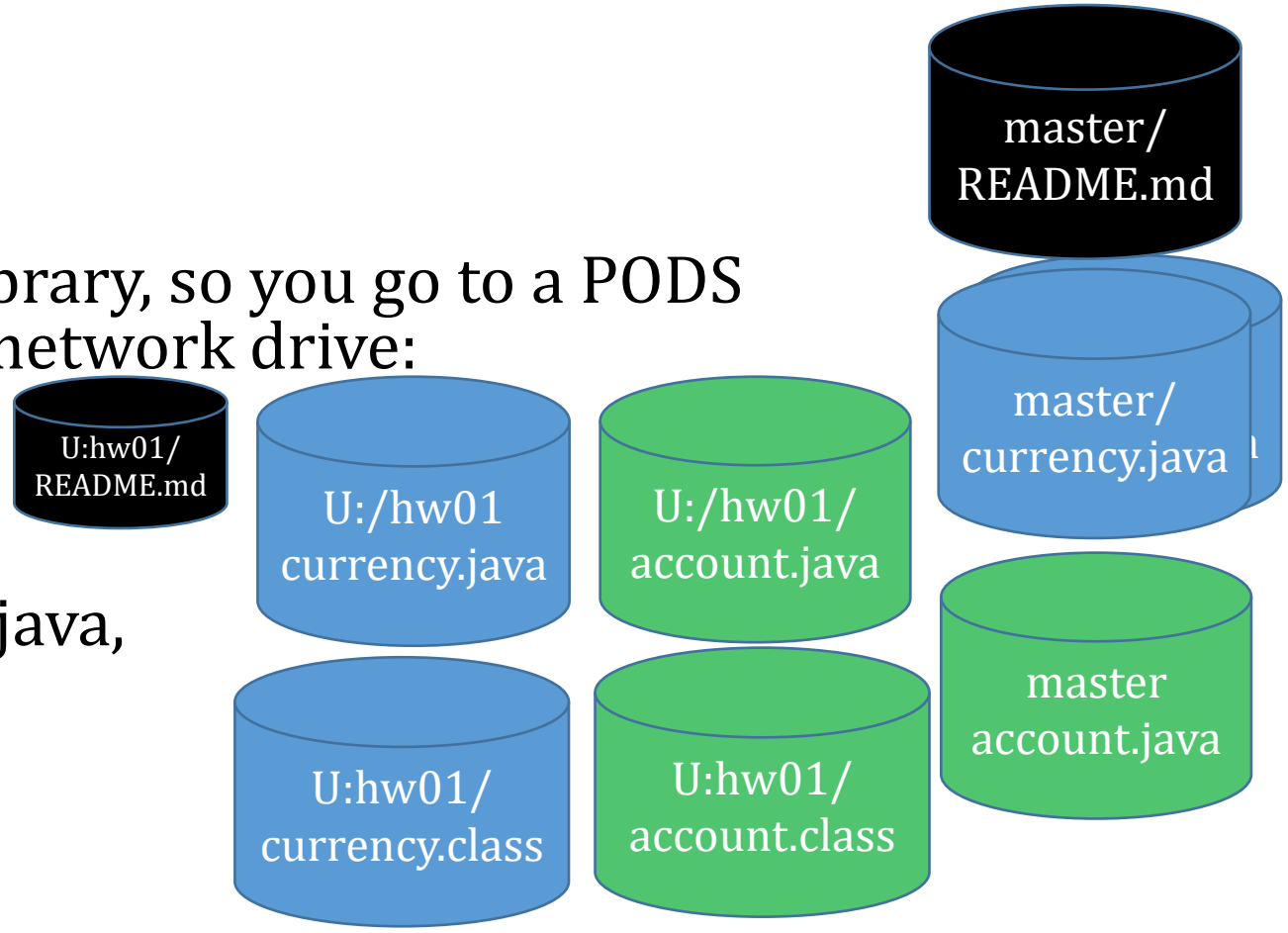




- When you are happy:
  git add currency.java
  git commit –a "added currency.java"
  git push

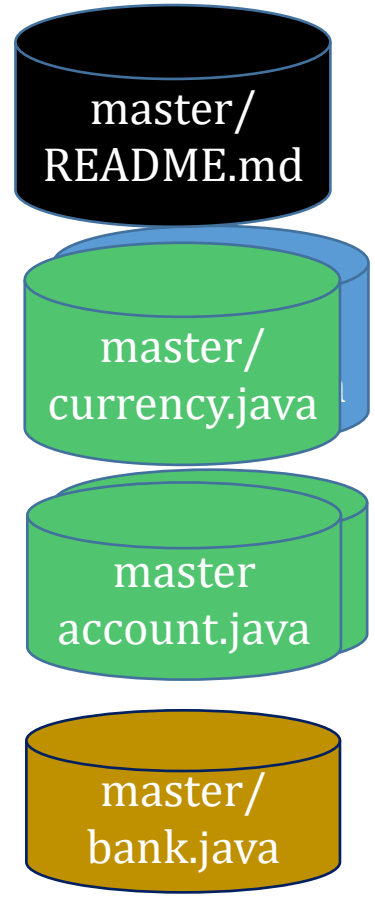


with git – 2

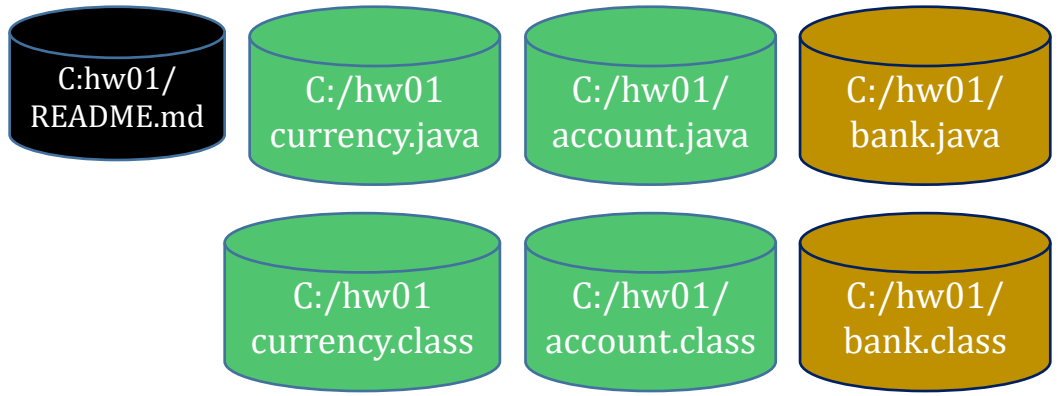


- You have some extra time in the library, so you go to a PODS machine, and clone hw01 to your network drive:
- And add code for the account:
- And compile to create:
- But you find mistakes in currency.java, so you need to update that file…
  - and recompile
- And commit and push:
  git add account.java
  git commit –a "Added account, fixed currency"
  git push

with git – 3



- Back in your apartment, you clone hw01to your laptop C drive.
- And add code for the bank:
- And compile to create:
- But you find mistakes in account.java, so you need to update that file…
  - and recompile
- And commit and push:
  git add bank.java
  git commit –a "Added bank, fixed account"
  git push



with git – 4

- The next day in the library, pull hw01 in your U: drive…
- And compile:
- Everything works! Turn in master!