

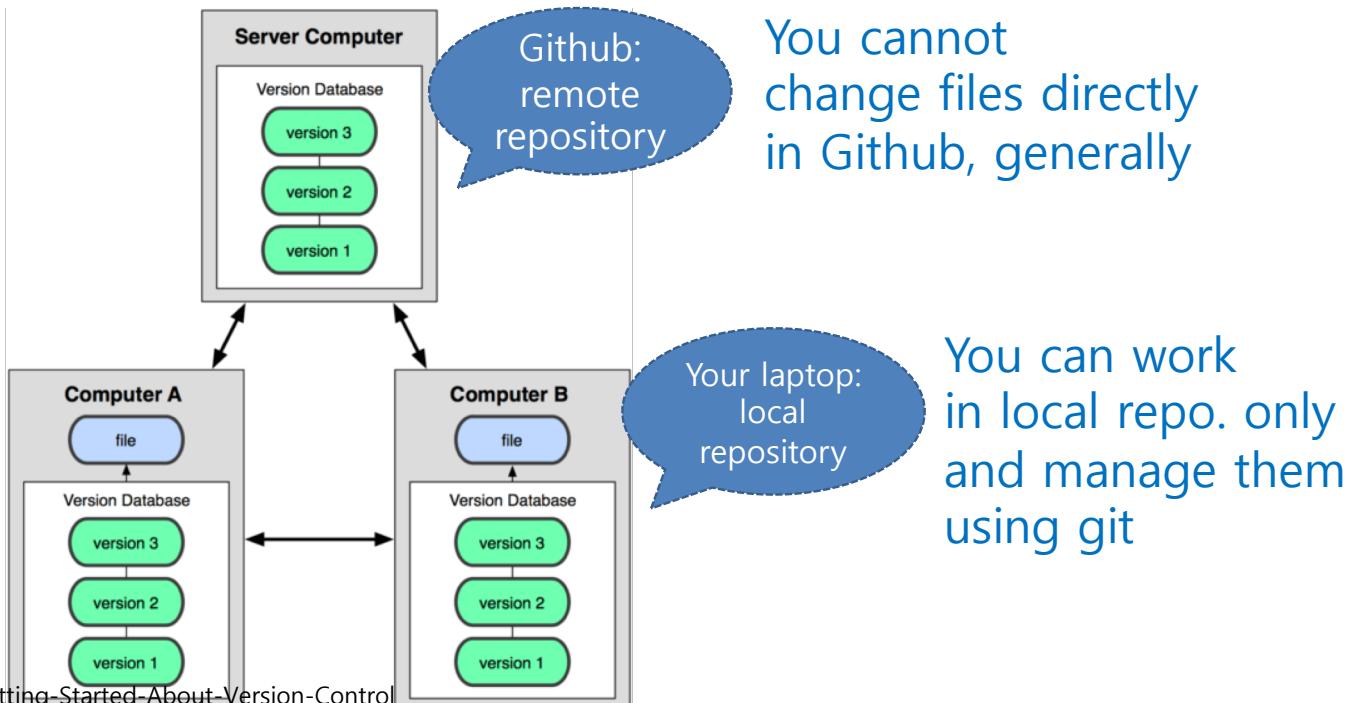
Git & Github for TinySE

Big Data Mining Lab.



What is Git and GitHub?

- **Github**: Web-based hosting service for version control using git
- **Git**: Difference to traditional version control systems (e.g., SVN and CSV)
 - Distributed version control and source code management



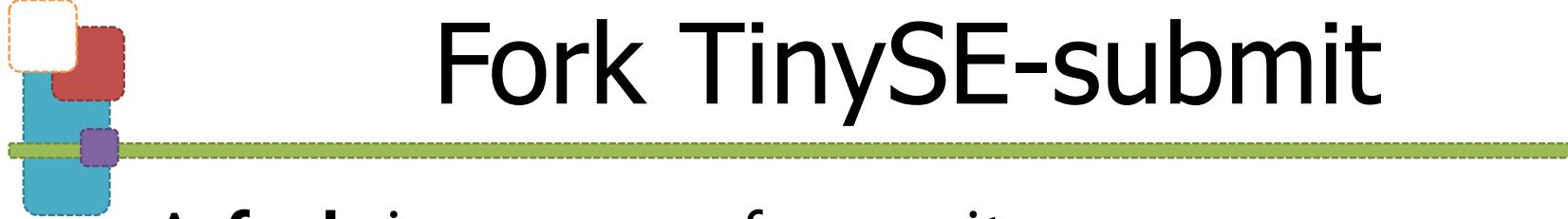


Installation (Git)

- Ubuntu
 - \$ sudo apt update
 - \$ sudo apt upgrade
 - \$ sudo apt install git
- Mac
 - Install the Xcode Command Line Tools
 - Or install using brew
- Then check your git version
 - \$ git --version

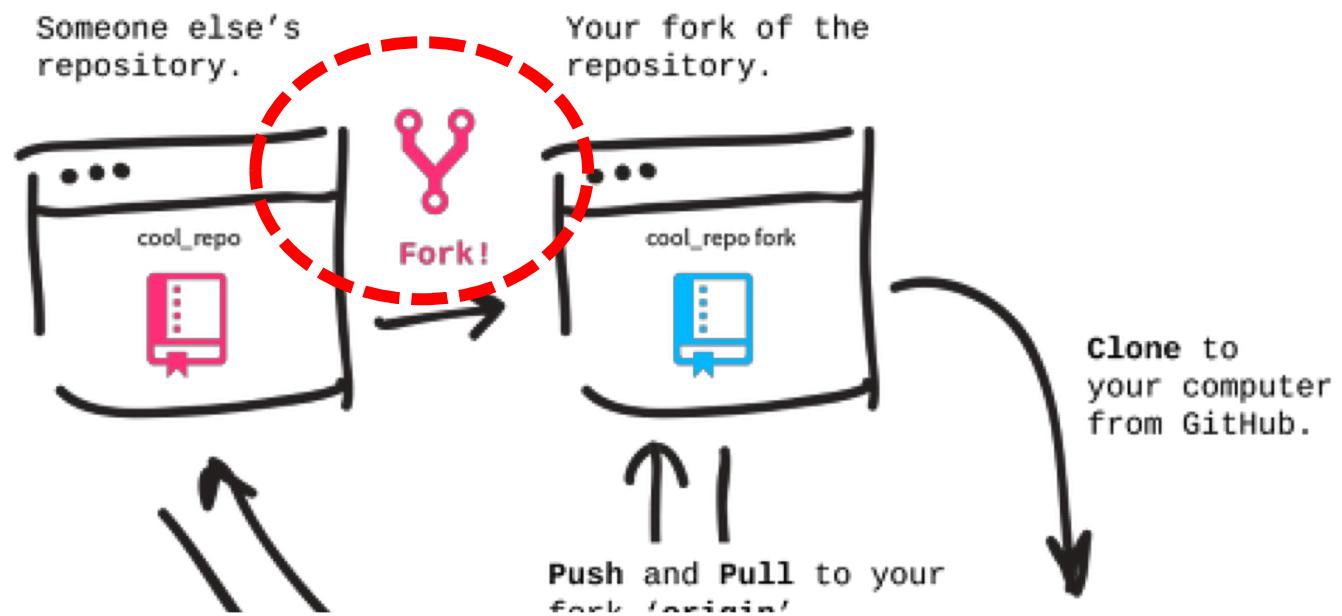
```
git version 2.17.1
```

FIRST STEP: FORK & CLONE

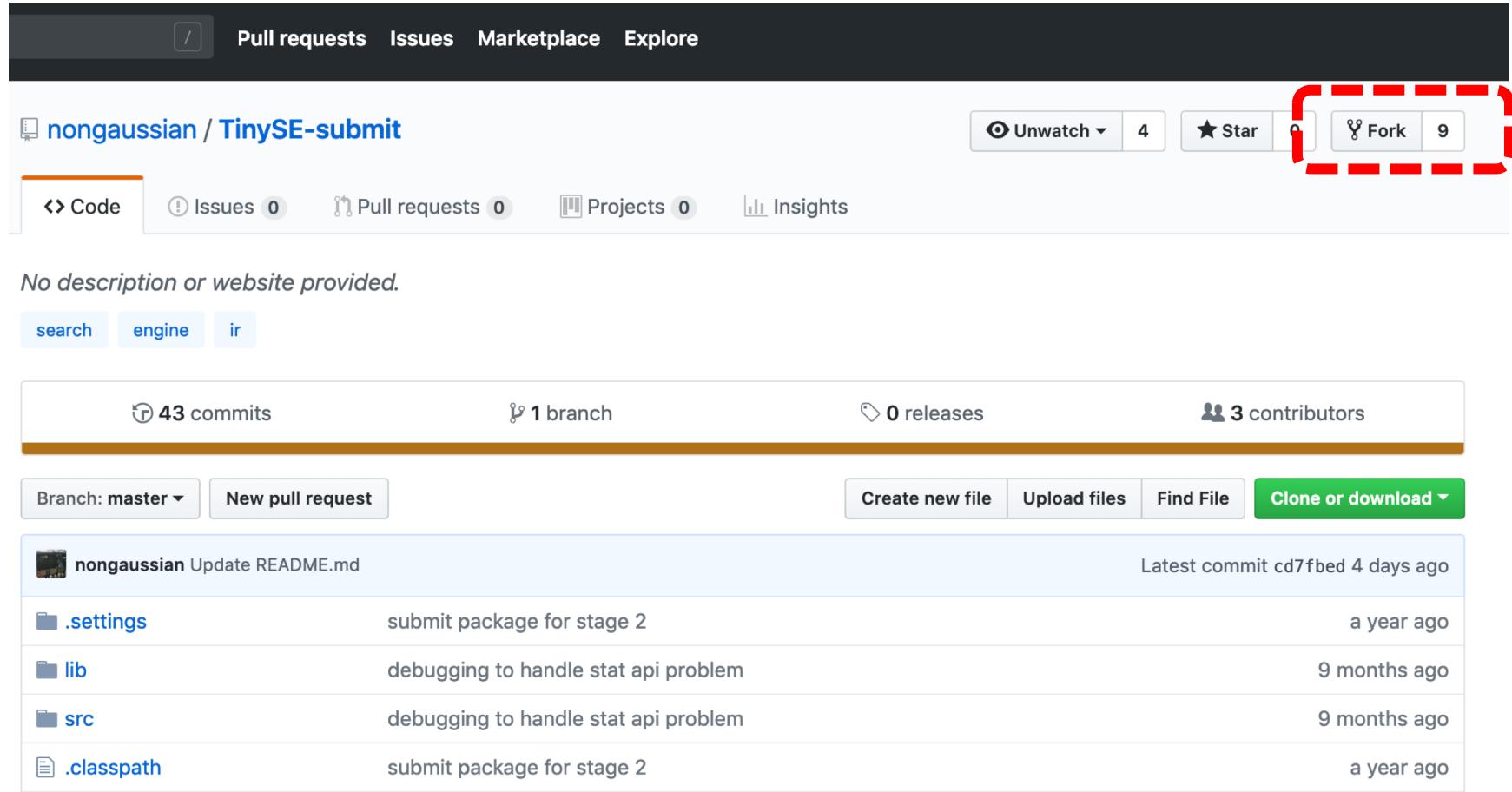


Fork TinySE-submit

- A **fork** is a copy of repository.
- **Forking** a repository allows you to freely experiment with changes without affecting the original project.



Fork TinySE-submit



Pull requests Issues Marketplace Explore

nongaussian / TinySE-submit

Unwatch 4 Star 0 Fork 9

Code Issues 0 Pull requests 0 Projects 0 Insights

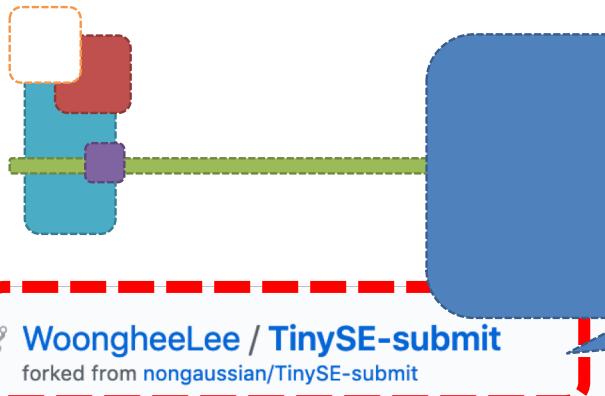
No description or website provided.

search engine ir

43 commits 1 branch 0 releases 3 contributors

Branch: master New pull request Create new file Upload files Find File Clone or download

File	Description	Time
README.md	Update README.md	Latest commit cd7fbcd 4 days ago
.settings	submit package for stage 2	a year ago
lib	debugging to handle stat api problem	9 months ago
src	debugging to handle stat api problem	9 months ago
.classpath	submit package for stage 2	a year ago



Then, you will have forked repository.

 WoongheeLee / [TinySE-submit](#)
forked from nongaussian/TinySE-submit

 Unwatch 1  Star 0  Fork 9

[Code](#) [Pull requests 0](#) [Projects 0](#) [Insights](#) [Settings](#)

No description, website, or topics provided. [Edit](#)

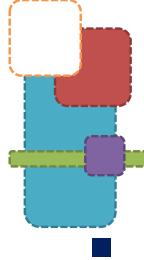
[Manage topics](#)

 43 commits  1 branch  0 releases  4 contributors

Branch: master [New pull request](#) [Create new file](#) [Upload files](#) [Find File](#) [Clone or download](#)

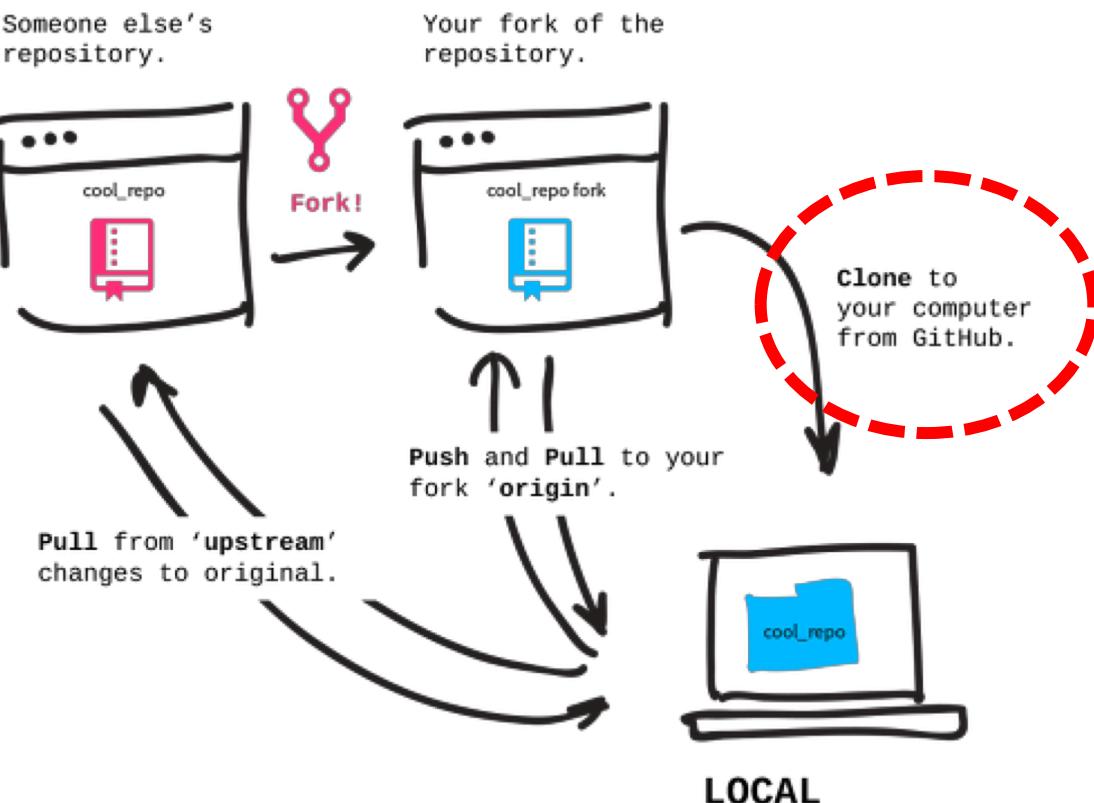
This branch is 1 commit ahead, 1 commit behind nongaussian:master. [Pull request](#) [Compare](#)

Author	Commit Message	Date
 WoongheeLee	Update README.md	Latest commit 27bd6ad 4 days ago
 .settings	submit package for stage 2	a year ago
 lib	debugging to handle stat api problem	9 months ago
 src	debugging to handle stat api problem	9 months ago
 __mainpath	submit package for stage 2	2 years ago



Clone Your TinySE-submit

- to create a clone, or copy of the target repository on the local computer



Use your computer's terminal to talk to two repositories via two remotes to the GitHub servers.

Clone Your TinySE-submit

The screenshot shows a GitHub repository page for 'WoongheeLee / TinySE-submit'. The repository was forked from 'nongaussian/TinySE-submit'. The page includes navigation tabs for Code, Pull requests (0), Projects (0), Insights, and Settings. A note states 'No description, website, or topics provided.' Below the repository summary, there are statistics: 48 commits, 1 branch, and 0 releases. A dropdown menu shows 'Branch: master' and a 'New pull request' button. On the right, there are buttons for Create new file, Upload files, Find File, and Clone or download (highlighted with a red dashed box). A modal window titled 'Clone with HTTPS' displays the URL <https://github.com/WoongheeLee/TinySE>. The page also shows commit history for 'master' and '.Classpath' files.

1. Click

2. Copy the repository url
(Team member copies leader's repo)

No description, website, or topics provided.

Manage topics

48 commits 1 branch 0 releases

Branch: master New pull request

Create new file Upload files Find File **Clone or download**

This branch is 5 commits ahead of nongaussian:master.

Clone with HTTPS Use SSH

Use Git or checkout with SVN using the web URL.
<https://github.com/WoongheeLee/TinySE>

Open in Desktop Download ZIP

a day ago

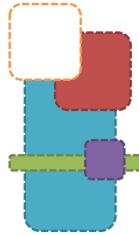
a year ago

10 months ago

.Classpath framework package of the 4th stage

citignore

framework package of the 4th stage



Clone Your TinySE-submit

- \$ git clone <your-forked-repository>
 - ex: \$ git clone https://github.com/WoongheeLee/TinySE-submit

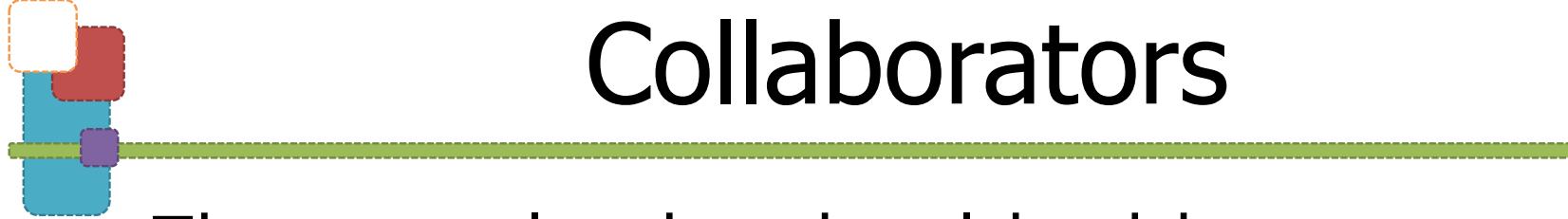
- \$cd TinySE-submit

BASIC CONCEPTS & COMMANDS



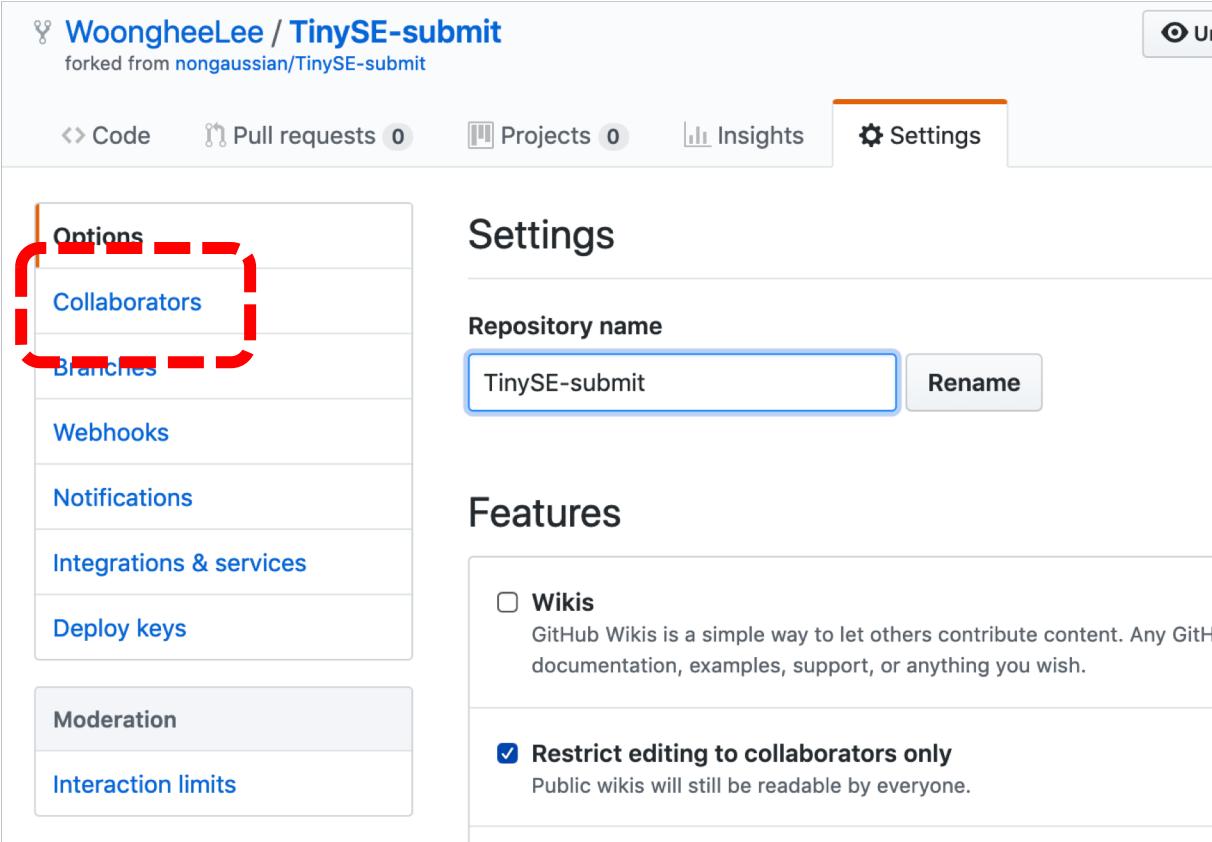
Add Collaborators

- An outside **collaborator** is a person who isn't explicitly a member of your organization, but who has **Read, Write, or Admin permissions** to one or more repositories in your organization.



Collaborators

- The team leader should add collaborators, if you have more members.



WoongheeLee / **TinySE-submit**
forked from nongaussian/TinySE-submit

Code Pull requests 0 Projects 0 Insights Settings

Settings

Repository name Rename

Features

Wikis
GitHub Wikis is a simple way to let others contribute content. Any GitHub documentation, examples, support, or anything you wish.

Restrict editing to collaborators only
Public wikis will still be readable by everyone.



Collaborators

- The team leader should add collaborators.

Collaborators

Push access to the repository

This repository doesn't have any collaborators yet. Use the form below to add a collaborator.

1. Search your team member

address

username instead.

kjwspecial

 kjwspecial

2. Add it

Add collaborator



Commit

- The "commit" command is used to save your changes **to the local repository**.
- Ex 1) change or add files
 - \$ git add file1 file name
 - \$ git commit -m "update file 1"
- Ex 2) Remove files
 - \$ git rm file2
 - \$ git commit -m "remove file 2"

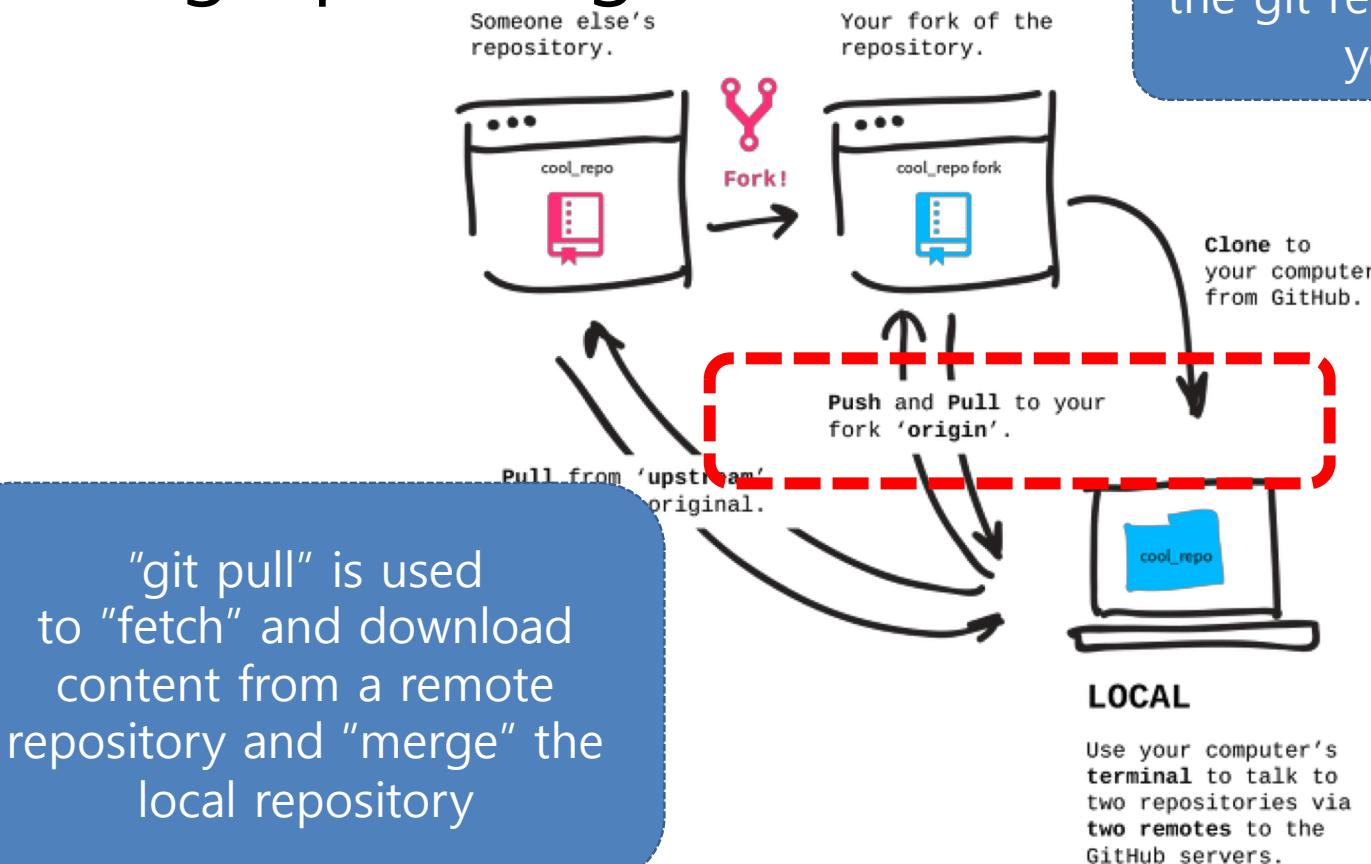
Push & Pull

Remote name

Branch name

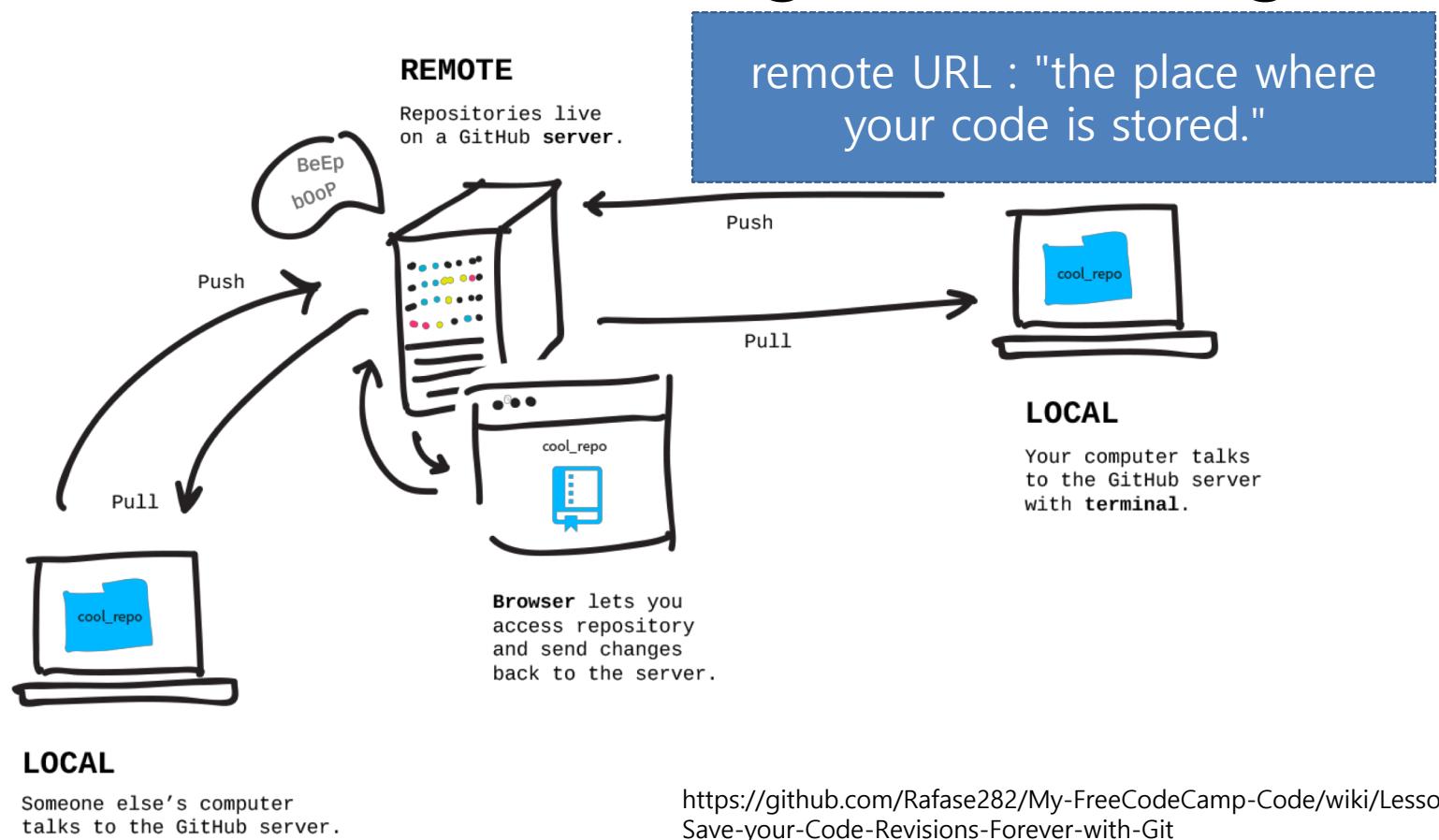
- \$ git push **[origin]** **[master]**
- \$ git pull origin master

"push" is used to upload on the git repository server what you "commit"



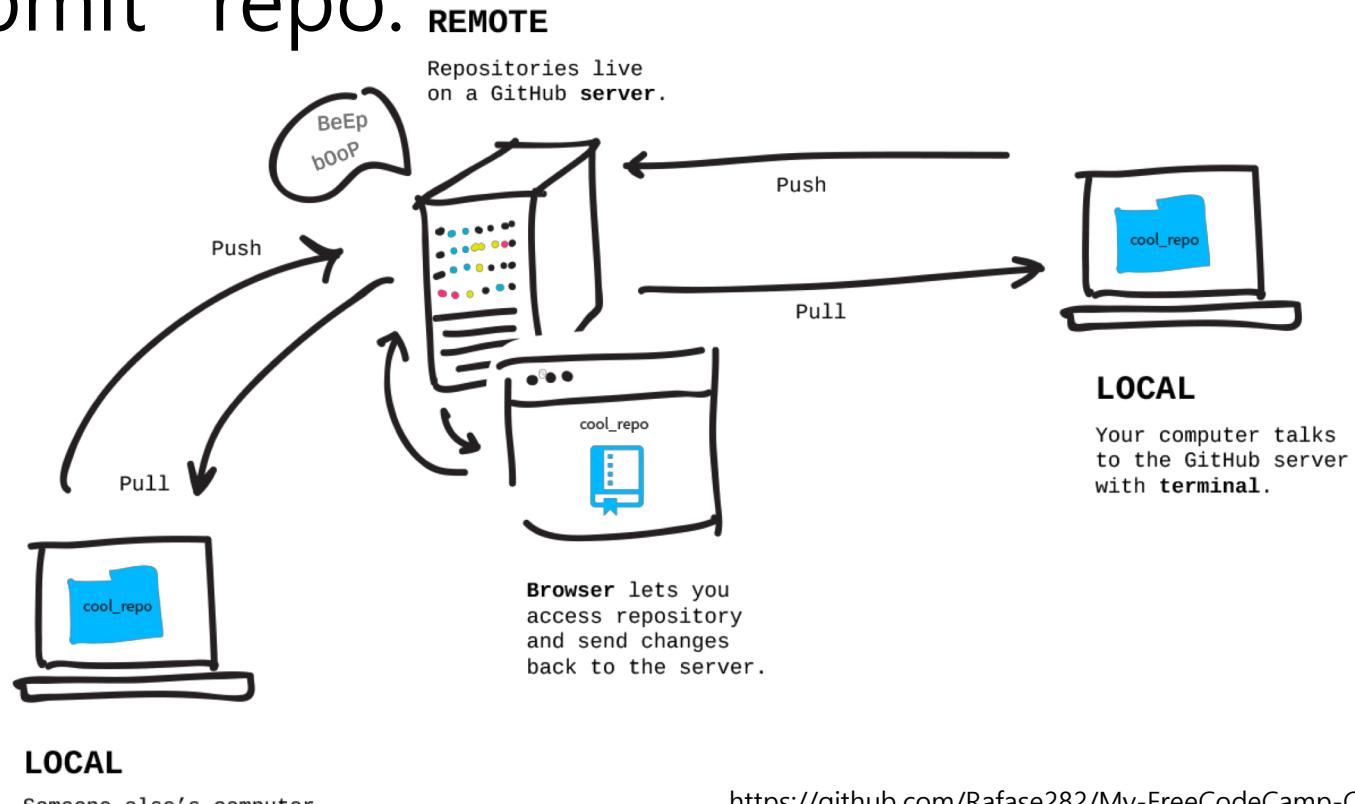
Remote

- a common repository that all team members use to exchange their changes



Remote

- In the class, students are NOT permitted to push on the “nongaussian/TinySE-submit” repo. REMOTE





Merge *upstream*'s changes into the local repository

- \$ git remote -v

"upstream" = original repo that you have forked

```
WoongheeLeeMacBookPro:TinySE-submit woonghee$ git remote -v
origin  https://github.com/WoongheeLee/TinySE-submit (fetch)
origin  https://github.com/WoongheeLee/TinySE-submit (push)
```

- Add remote
 - \$ git remote add <remote-name> <upstream-url>

```
git remote add upstream https://github.com/nongaussian/TinySE-submit
```



Merge *upstream*'s changes into the local repository

- Check current remote list
 - \$ git remote -v

```
WoongheeLeeMacBookPro:TinySE-submit woonghee$ git remote -v
origin  https://github.com/WoongheeLee/TinySE-submit (fetch)
origin  https://github.com/WoongheeLee/TinySE-submit (push)
upstream  https://github.com/nongaussian/TinySE-submit (fetch)
upstream  https://github.com/nongaussian/TinySE-submit (push)
```

New remote named "upstream" is added.

- Fetch upstream

It fetches all branches of "upstream".

- \$ git fetch upstream

```
WoongheeLeeMacBookPro:TinySE-submit woonghee$ git fetch upstream
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 2), reused 1 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/nongaussian/TinySE-submit
 * [new branch]      master    -> upstream/master
```



Merge *upstream*'s changes into the local repository

- Check out (to be local master branch)
 - \$ git checkout master

```
[WoongheeLeeMacBookPro:TinySE-submit woonghee$ git checkout master
 Already on 'master'
 Your branch is up to date with 'origin/master'.
```

- Merge “upstream/master” with “master(student’s local branch)”
 - \$ git merge upstream/master

Branch name



How To Handle Conflicts

Scenario:

- You forked & clone TinySE-submit
- You change “README.md” file on your repository & commit

```
1 # Team
2 -----
3 * WoongheeLee 20000000XXXX
4 * JongwooKim 20000XXXXX
5 -----
6 # How to submit your project output
7 1. Open <code>pom.xml</code> and c
```

```
[WoongheeLeeMacBookPro:TinySE-submit woonghee$ git add README.md
[WoongheeLeeMacBookPro:TinySE-submit woonghee$ git commit -m "student ID"
```

- TA changes “README.md” on the upstream repository
- You try to merge upstream into your local

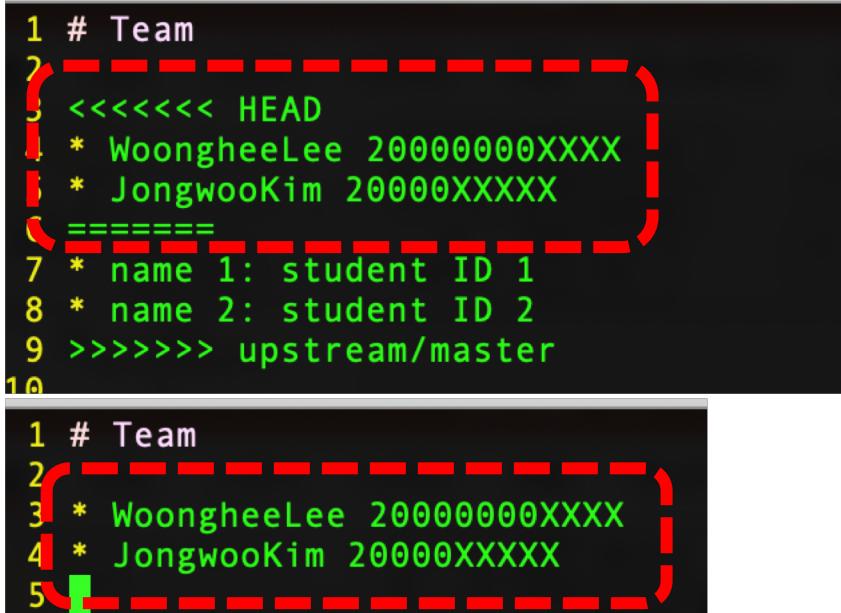
```
[WoongheeLeeMacBookPro:TinySE-submit woonghee$ git merge upstream/master
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
Automatic merge failed; fix conflicts and then commit the result.
```



How To Handle Conflicts

- Open the conflict file with text editor

```
[WoongheeLeeMacBookPro:TinySE-submit woonghee$ vi README.md]
```



```
1 # Team
2
3 <<<<< HEAD
4 * WoongheeLee 20000000XXXX
4 * JongwooKim 20000XXXXX
5 =====
6 * name 1: student ID 1
7 * name 2: student ID 2
8 >>>>> upstream/master
9
10
1 # Team
2
3 * WoongheeLee 20000000XXXX
4 * JongwooKim 20000XXXXX
5
```



Check every single conflicts

- Add and commit it

```
[WoongheeLeeMacBookPro:TinySE-submit woonghee$ git add README.md]
```

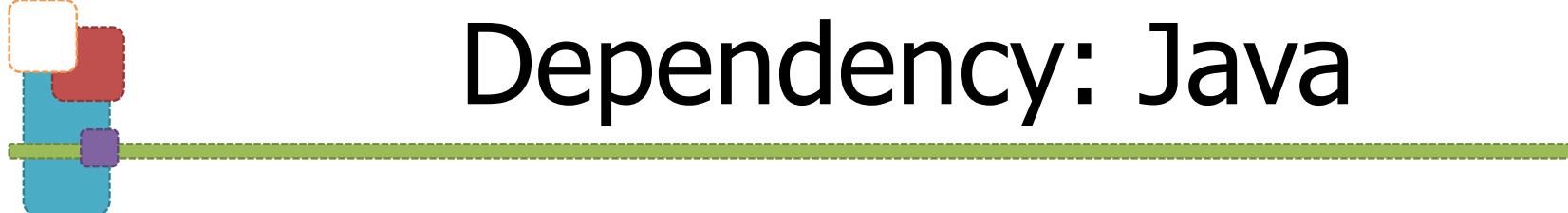
```
[WoongheeLeeMacBookPro:TinySE-submit woonghee$ git commit -m 'merge with student id'
[master 3d3e513] merge with student id
```

MAVEN



Maven

- A **software project management** tool
- Based on the concept of a project object model (POM),
 - Maven can manage a project's build, package dependencies, reporting and documentation from a central piece of information.



Dependency: Java

- Installing Java (Ubuntu)
 - \$ sudo add-apt-repository
ppa:webupd8team/java
 - \$ sudo apt-get update
 - \$ sudo apt-get install oracle-java8-installer
 - \$ java -version

```
$ java -version
java version "1.8.0_201"
Java(TM) SE Runtime Environment (build 1.8.0_201-b09)
Java HotSpot(TM) 64-Bit Server VM (build 25.201-b09, mixed mode)
```



Maven Installation (Ubuntu)

- sudo apt-get install maven

```
$ sudo apt-get install maven
```

```
Reading package lists... Done  
Building dependency tree  
Reading state information... Done
```

```
Setting up libmaven3-core-java (3.5.2-2) ...  
Setting up maven (3.5.2-2) ...  
update-alternatives: using /usr/share/maven/bin/mvn to provide /usr/bin/  
mvn (mvn) in auto mode
```

- mvn --version

```
$ mvn --version
```

```
Apache Maven 3.5.2  
Maven home: /usr/share/maven  
Java version: 1.8.0_201, vendor: Oracle Corporation  
Java home: /usr/lib/jvm/java-8-oracle/jre  
Default locale: en_US, platform encoding: UTF-8  
OS name: "linux", version: "4.15.0-46-generic", arch: "amd64", family: "unix"
```

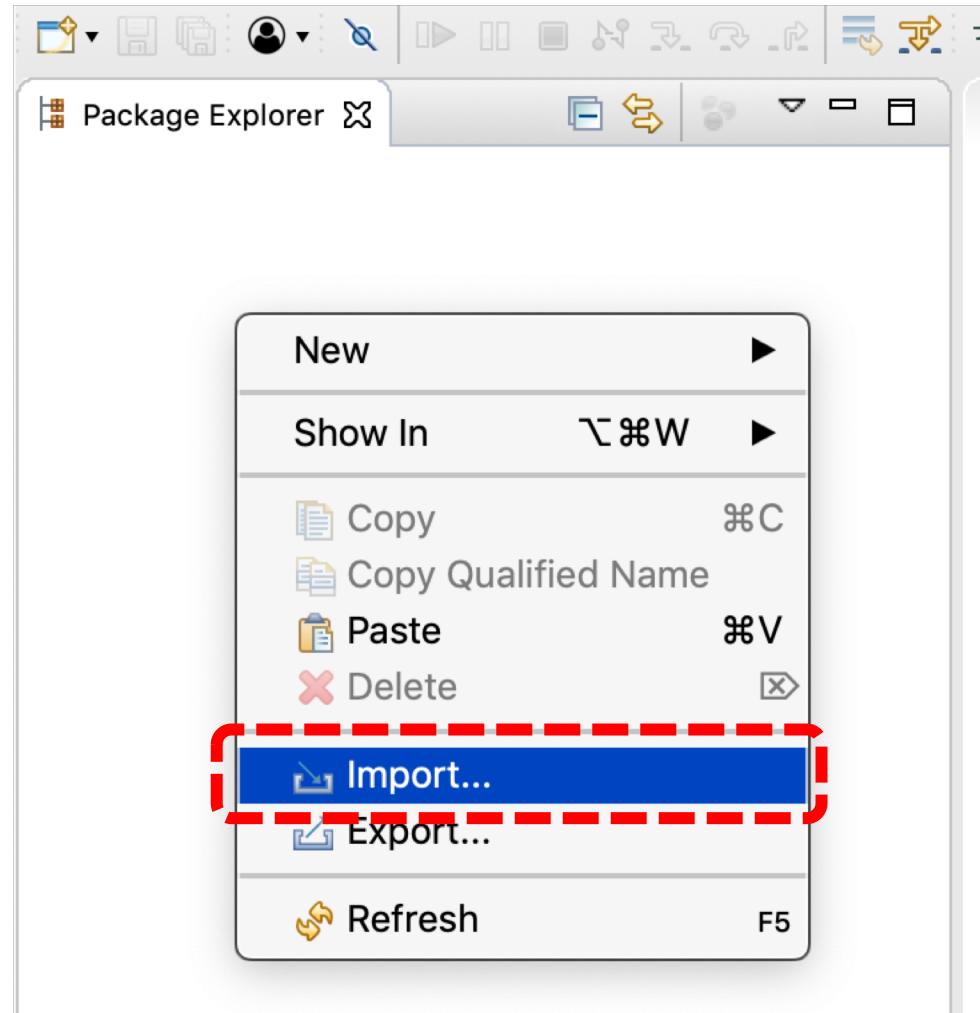


Installation (MAC)

- 1. use brew
- 2. or install with binary file
 - Download apache-maven-<version>.zip file from
<http://maven.apache.org/download.cgi>
 - Unzip the file
 - Copy the dir. to /usr/local/ (or /opt)
 - Set \$PATH
 - \$ vi ~/.bash_profile
 - export PATH= ... :/usr/local/apache-maven-<version>/bin
 - \$ source ~/.bash_profile
 - \$ mvn -v
 - Check \$JAVA_HOME
 - export JAVA_HOME=/Library/Java/JavaVirtualMachine/jdk<version>.jdk/Contents/Home/



Importing Maven Project into Eclipse



Importing Maven Project into Eclipse (cont.)

Select

Import Existing Maven Projects



Select an import wizard:

type filter text

- ▶ General
- ▶ Git
- ▶ Gradle
- ▶ Install
- ▶ Maven

Check out Maven Projects from SCM

Existing Maven Projects

Install or deploy an artifact to a Maven repository

Materialize Maven Binary Project

Materialize Maven Projects from SCM

- ▶ Oomph

- ▶ Run/Debug

- ▶ Tasks

- ▶ Team

- ▶ XML

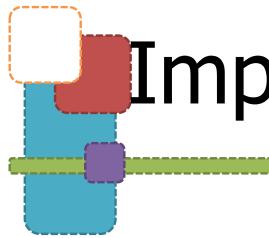


< Back

Next >

Cancel

Finish



Importing Maven Project into Eclipse (cont.)

Maven Projects
Select Maven projects

Root Directory: /Users/woonghee/Documents/git/TinySE-submit

Projects:

- /pom.xml edu.hanyang:2019123456:0.0.1-SNAPSHOT:jar

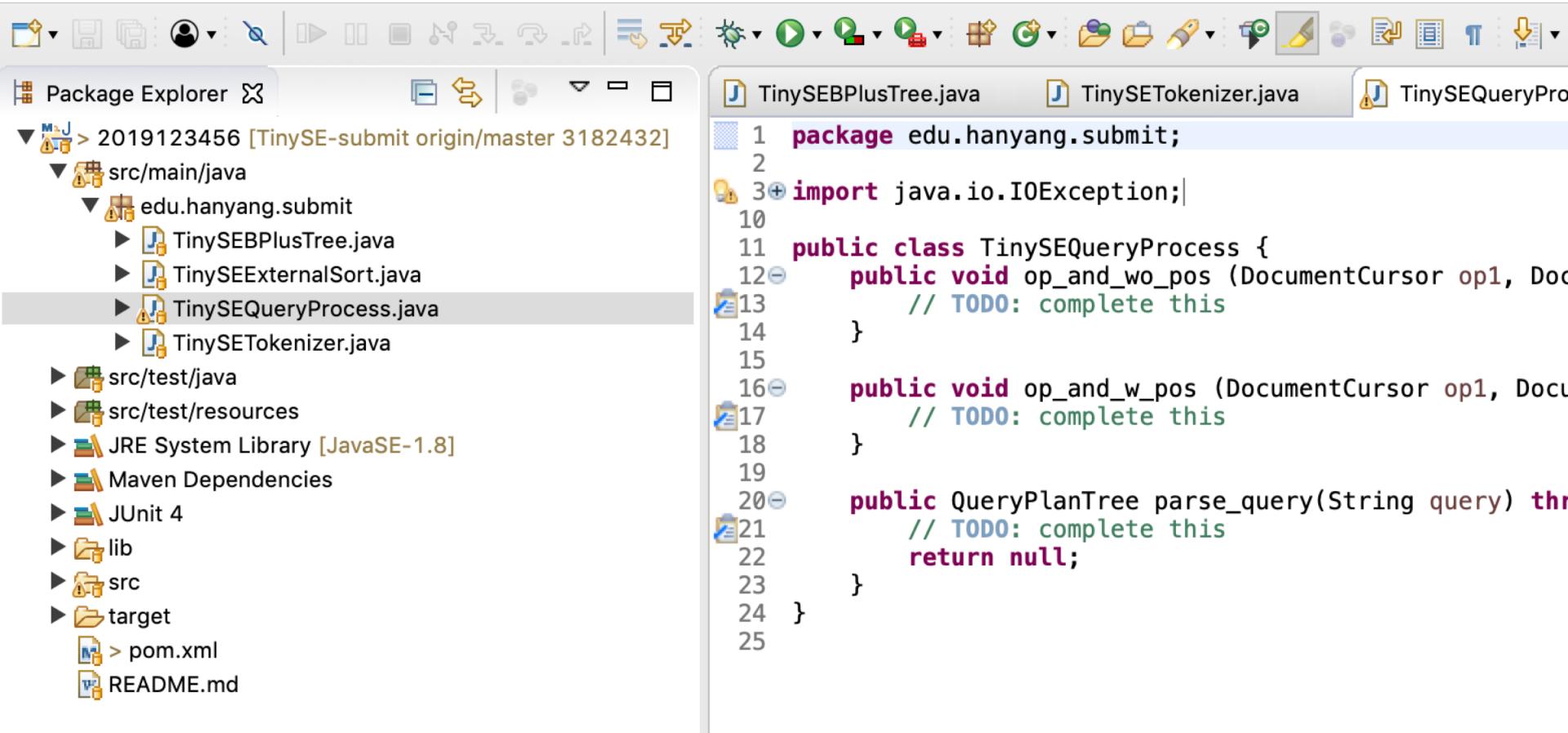
Add project(s) to working set
2019123456

► Advanced

Git clone directory



Importing Maven Project into Eclipse (cont.)



The screenshot shows the Eclipse IDE interface with the following details:

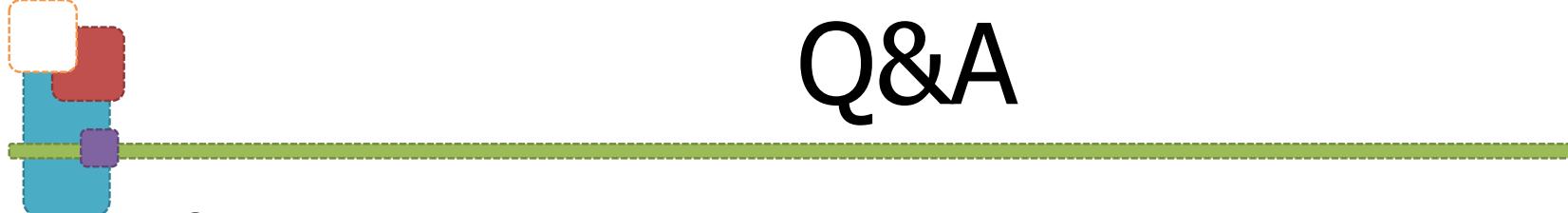
- Toolbar:** Standard Eclipse toolbar with icons for file operations, search, and project management.
- Package Explorer:** Left-hand view showing the project structure:
 - Project: 2019123456 [TinySE-submit origin/master 3182432]
 - src/main/java:
 - edu.hanyang.submit:
 - TinySEBPlusTree.java
 - TinySEExternalSort.java
 - TinySEQueryProcess.java** (highlighted)
 - TinySETokenizer.java
 - src/test/java
 - src/test/resources
 - JRE System Library [JavaSE-1.8]
 - Maven Dependencies
 - JUnit 4
 - lib
 - src
 - target
 - pom.xml
 - README.md
- Code Editor:** Right-hand view showing the content of **TinySEQueryProcess.java**:

```
1 package edu.hanyang.submit;
2
3+ import java.io.IOException;
10
11 public class TinySEQueryProcess {
12     public void op_and_wo_pos (DocumentCursor op1, DocumentCursor op2)
13         // TODO: complete this
14     }
15
16     public void op_and_w_pos (DocumentCursor op1, DocumentCursor op2)
17         // TODO: complete this
18     }
19
20     public QueryPlanTree parse_query(String query) throws IOException
21         // TODO: complete this
22         return null;
23     }
24 }
25 }
```



mvn package, mvn test

- mvn package
 - Take the compiled code and package it in its distributable format, such as a JAR
 - To submit the result of each stage
- mvn test
 - During development, you may run a single test class repeatedly. To run this through Maven, set the test property to a specific test case.
 - To test your code before submit



Q&A

- If you have any question, please visit TA during office hour.
 - TA: Woonghee Lee
 - woongheelee@hanyang.ac.kr
 - Room 314, 4-th engineering building
 - Office hour: Mon. 16:00 ~ 17:00