Using Github

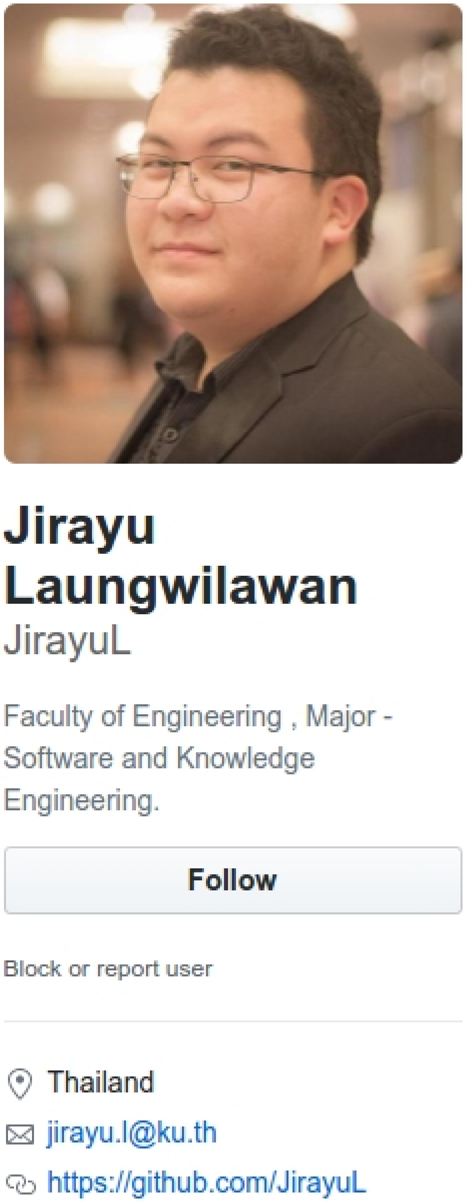
What Github Does

- Online project hosting site
- Share git repositories, with access control
- Issue Tracking
- Project Boards
- Automated testing, builds, & other services
- Documentation wiki and web pages (github.io)
- Integrates with other services, e.g. Travis CI

Github Profile

Example of SKE student profiles.

1. Real name
2. Photo
3. (Optional) Email
4. Description of you



Jirayu Laungwilawan

JirayuL

Faculty of Engineering , Major - Software and Knowledge Engineering.

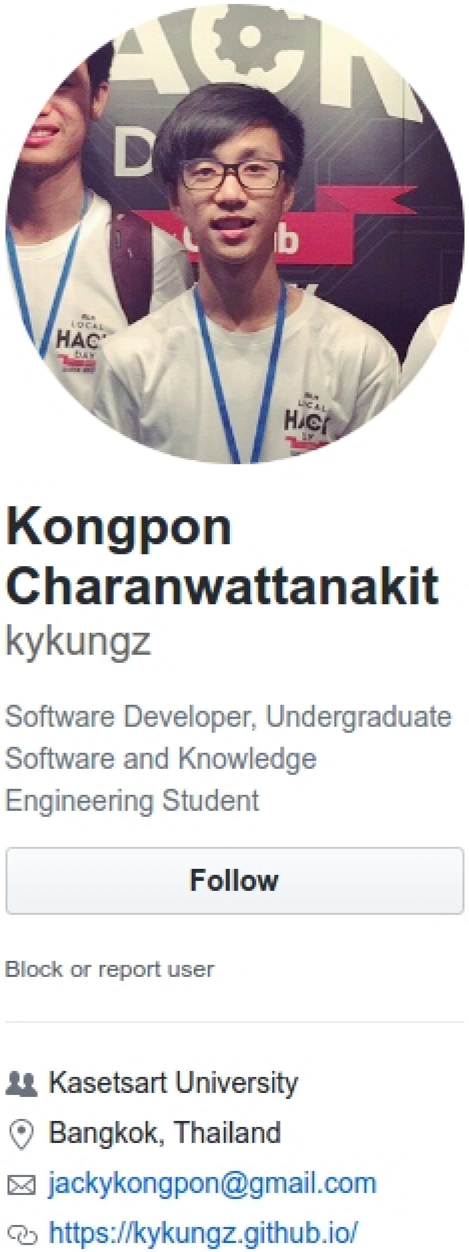
Follow

Block or report user

Thailand

jirayu.l@ku.th

https://github.com/JirayuL



Kongpon Charanwattanakit

kykungz

Software Developer, Undergraduate Software and Knowledge Engineering Student

Follow

Block or report user

Kasetsart University

Bangkok, Thailand

jackykongpon@gmail.com

https://kykungz.github.io/

Creating and using a Repository

Case 1: *Project code is on your local computer.*
You want to copy it to Github.

Case 2: *Project already exists on Github.*
You want to copy it to your computer.

Special Case:

Case 3: *A new project (no files yet).*

Case 1: Starting from Local Project

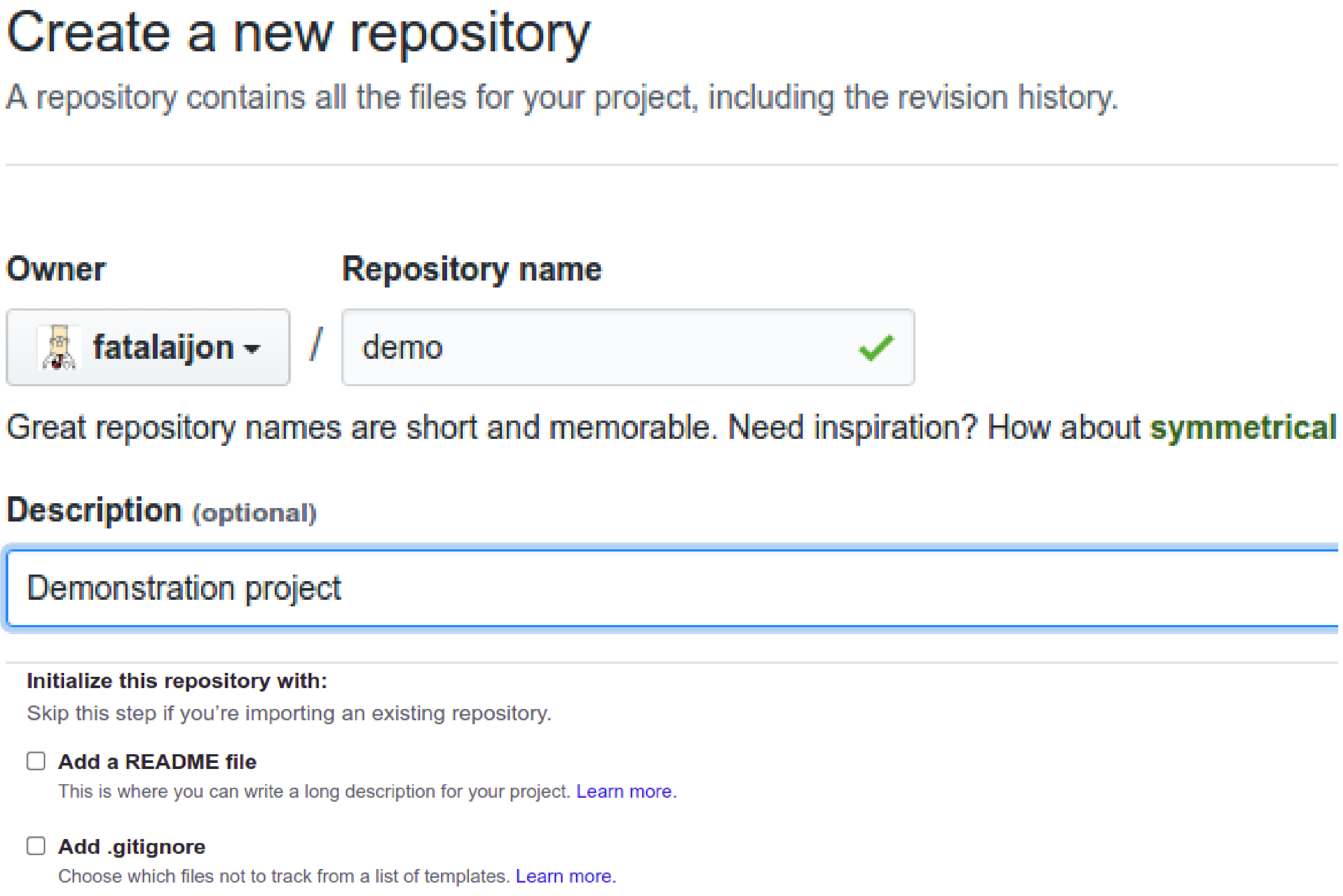
You already have a project on your computer

1. Create a local "git" repository.

```
cmd> git init
# These two files are typical
cmd> git add .gitignore README.md
# Add some source code
cmd> git add src/*.java (for example)
# Commit code to github
cmd> git commit -m "initial code checkin"
```

Case 1: Remote must be empty

2. On Github, create an **empty** repository.



Create a new repository

A repository contains all the files for your project, including the revision history.

Owner: fatalaijon ▾

Repository name: demo ✓

Great repository names are short and memorable. Need inspiration? How about **symmetrical**

Description (optional): Demonstration project

Initialize this repository with:

Skip this step if you're importing an existing repository.

□ **Add a README file**

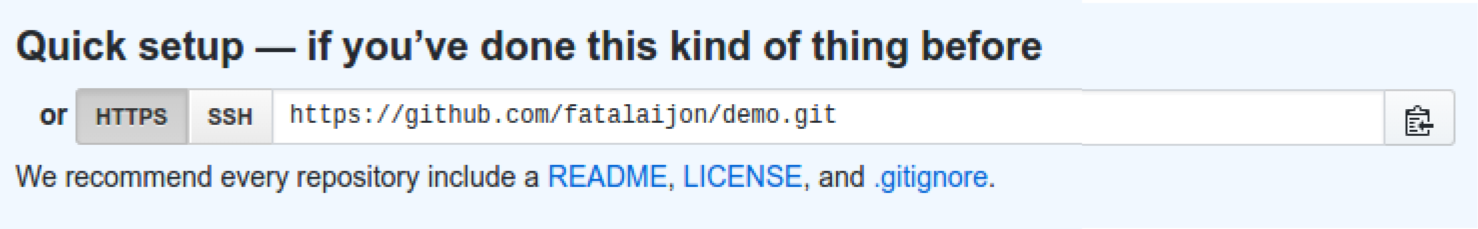
This is where you can write a long description for your project. Learn more.

□ **Add .gitignore**

Choose which files not to track from a list of templates. Learn more.

Case 1: add Github as remote

3. Copy the URL of new Github repository (https or ssh).



Quick setup — if you've done this kind of thing before

https://github.com/fatalaijon/demo.git

We recommend every repository include a README, LICENSE, and .gitignore.

4. In your local project, add Github as a remote repository named "**origin**":

```
cmd> git remote add origin
    https://github.com/fatalaijon/demo.git
```

Case 1: push your code to Github!

5. Push (copy) the local repository to Github

```
cmd> git push -u origin master
```

You only need **"-u origin master"** the **first time** you push to Github.

Next time, just type "**`git push`**".

"master" or "main" ?

The standard name of the default git branch is **`master`**

But some people object to the words "master" and "slave".

So, Microsoft changed the name to **`main`**

You can use either name (master or main).

Github quietly tries to get you to rename your default branch to "main" by offering this cut & paste code:

…or push an existing repository from the command line

```
git remote add origin git@github.com:fatalaijon/demo.git

git branch -M main

git push -u origin main
```

Be careful what you copy and paste!

Case 2: Starting from Github

A project already exists on Github.
You want to "clone" it your local computer.

1. On Github, copy the Github project URL

```
https://github.com/user/demo.git
```

or: go to project on Github and click on

and copy the URL.

2. In your workspace directory, type:

```
cmd> git clone https://github.com/user/demo
```

NOTE: "git clone" creates a <u>new</u> directory named "demo" inside your current directory. If this directory already exists, clone won't work.

Case 2: ready to use

That's it!

Github is automatically the remote named "`origin`".

To copy your local work back to Github, just "**`git push`**" your commited work.

Case 2a: use a different project name

The name of your **local project directory** (directory name) can be **different** from the Github repository name.

1) Specify a local directory name when you "**clone**":

```
# Clone "demo" into local directory "mydemo"

cmd> git clone https://github.com/fatalaijon/demo.git  mydemo
```

Syntax: `git clone` *`remote_url local_repo_name`*

-- or --

2) rename the directory yourself!

use any file manager to rename directory

Comparison of 2 Cases

(*done in class*)

Case 3: You don't have a project yet

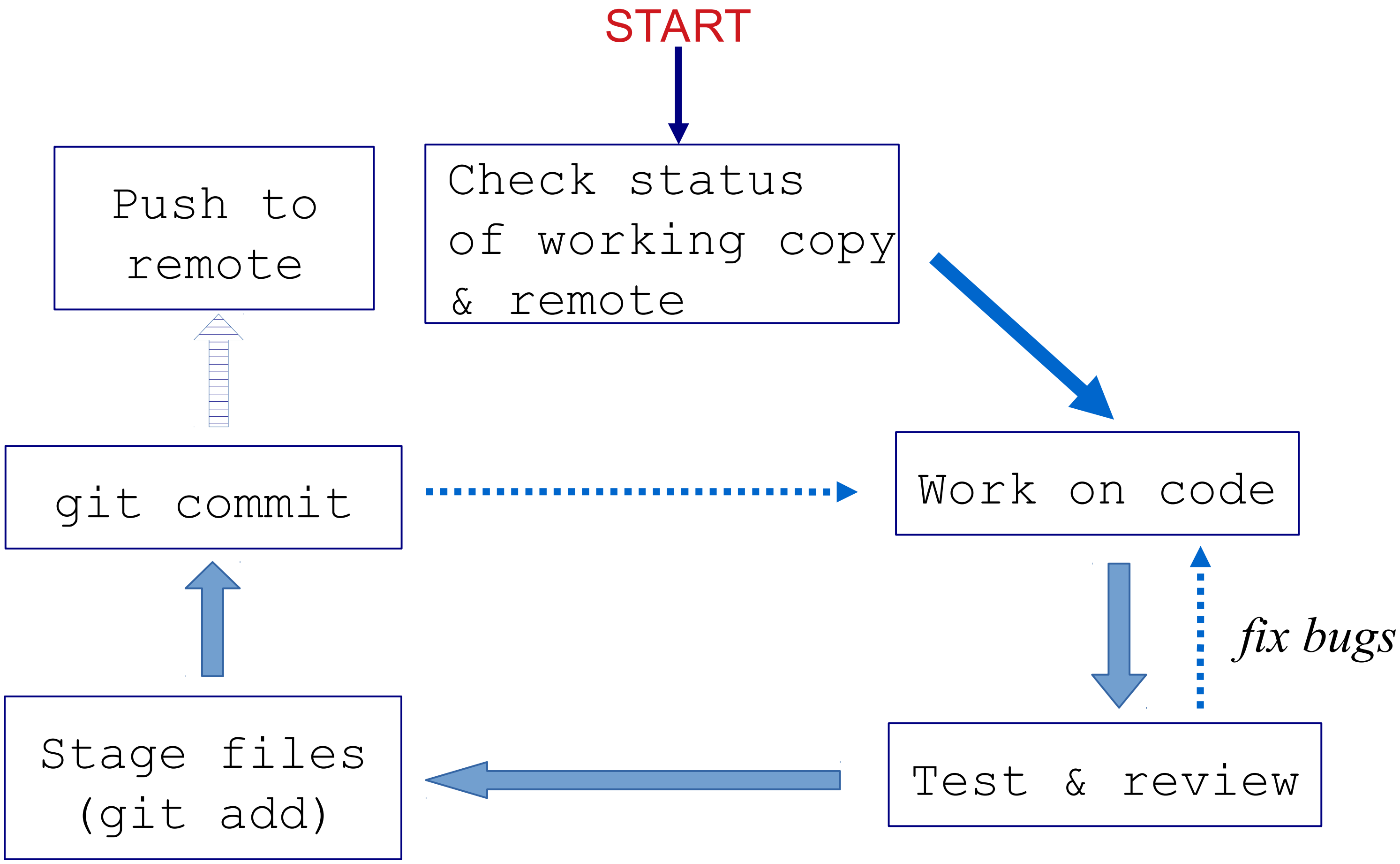
You can use either case 1 or case 2.

Case 1: Create a local project first. Seems more natural.

Case 2: Create a new project on Github with README and .gitingore, then clone it. Requires less typing.

Pro Case: IDEs can do some of this automatically. But you need to understand the basic way first.

Workflow for an *individual* project



Git Workflow for an *Individual* project

1) Check status of your working copy (*)

```
cmd> git status
```

It should be clean. If not, do "git diff" and then...

2) Commit changes <u>or</u> update your working copy.

```
(git diff, git add -u, git commit)
```

3) Do some work:

Code, test. Code, test. Review.

(*) *if you work on more than one computer, you need to "fetch" or "pull" any work from Github that is not on <u>this</u> computer (i.e. this local repo).*

Git Workflow (cont'd)

4) After code-test-reivew: check status again

```
   cmd> git status
   Changes not staged for commit:
      modified: src/Problem2.java
   Untracked files:
      src/Problem3.java
```

5) Add and commit your work to the local repository

```
cmd> git add src/Problem2.java src/Problem3.java
cmd> git commit -m "Solved problems 2 and 3"
[master 29abae0] Solved problem 2 and 3
2 files changed, 44 insertions(+), 5 deletions
```

Git Workflow (update remote)

6) Push the changes to Github

```
cmd> git push
Compressing objects: 100% (12/12), done.
Writing objects: 100% (12/12), 3.60 KiB,
done.
Total 12 (delta 9), reused 0 (delta 0)
remote: Resolving deltas: 100% (9/9), ...
To https://github.com/fatailaijon/demo.git
 468abdf..29abae0  master -> master
```

7) Take a break.

That's it!   Repeat the cycle as you work.

Github Workflow for Team Projects

On a **team project**, other people will commit files to the **same** Github repository!

You should update your local repository from Github before you try to "push" your work to Github.

Use "Github Flow" as workflow in team projects.

"Github Flow" is a separate topic in this course. It is good for both team and solo projects.

Github Flow is the convention for team work in this course.

Github Classroom

Github Classroom automates creating a git project with starter code (template code).

The steps are:

1. Instructor provides a URL for a Github assignment.

2. You visit the URL. Login to Github if necessary.

3. "Accept" the assignment.

4. Github creates a repo with starter code for you. It offers a link to the repo. It *may* redirect you there.

Github Classroom (2)

5. The README file in your repo may contain instructors. Read it.

6. Clone the repo in the usual way.

7. Do the assignment on your local computer.

8. Important: Commit all your work. Don't forget to add any new files.

9. "git push" to push to Github.

10. Check your repo on Github. Is your work there?