**GitHub Class**

Git is a tool for tracking the files’ changes and working together on the same file within a group. We will use Github for now and for the future projects.

There are two ways to use GitHub, one is use the GitHub Desktop and GitHub Page; another way, a simpler way is using command line. By using the GitHub Desktop Application, we can achieve about 80% of the total features of Github or Git. If you want to learn how to use command line to do the following thing or you want to transfer your files between repositories, you can either search them or ask me. Or you meet some problems cannot be solved by the GUI, you can learn how to use command line via reading Git Pro. I have include that book in our Repository.

After this one hour class, if you want to have a test, you can use our GitHub Class Repository, also you and download that book.

In this manual, I only include the basic skills of how to use the GitHub to work on the same code with your partner. All the contents with underline refer to the GitHub Page; otherwise refer to GitHub Desktop Application

* Repository

It’s like a folder where you put all the files in it, and everyone can download the updated files from it and upload changes to it. Only the owner of the repository can approve the merge request. We will talk about the details later.

* Branch
  + Master

**Do not directly work on the Master branch**

* + - Remote branch
    - Local branch
  + Other

If you want to work one part of the entire code, you are supposed to create a separate branch for yourself or for your small group. When you follow the steps of creating a new branch (either a local branch, remote branch or both) you will have the same thing as the master branch before you start your work. Whatever you edit on your branch will not change the master branch unless owner confirms your merge request.

* + - Remote branch
    - Local branch
* Create a New branch
  + From Desktop
    - Update your local master
      1. Change your current branch to master
      2. Repository---**Pull**
    - Branch—New Branch

The branch created above is only a local branch, will not show up on the GitHub Page

* + - Repository---**Push**

Then this branch will become a remote branch

* + From GitHub Page
    - Code---Branch :master
    - Enter the name of the new branch
      1. If you want the new branch has the same thing as Master, you have to activate your master branch before you create the new branch
      2. The branch created above is a remote branch, and it can’t be found from your local
    - Desktop---Repository---**Pull**
  + Download a existing remote branch to your local
    - Desktop---Repository---**Pull**
* Download from Remote master to your local branch (**Before you start to work**)
  + If you want to create a new branch, follow the above steps to do so.
  + If you want to use the existing local branch

1. Change the Current branch to your local branch
2. Branch—Merge Into current branch—Merge
3. **Push**

Then you can work on your own branch

* Upload your local branch to your remote branch (**After you complete all your changes**)
  + Complete your job, save it.

You can find all your change from GitHub Desktop Application

* + - 1. With “+” means you add this line
      2. With “-“ means you delete this line
      3. Make a commit to your change, so that your partner can understand why you made such changes. You can commit on the left-down corner.
      4. Click Commit to XXX(Your branch name)
      5. Click **Push origin**
  + Open GitHub Page

1. **Pull requests**
2. Compare your branch to master
3. Base: master, compare: Your Branch
4. **Create Pull request**
   * Owner of this repository has the right to confirm your request

**Caution: If more than one person is working on the same section of the code, you might have conflicts when you try to merge both of your changes to master. You have to solve the conflict before owner can confirm all the requests.**

1. Open **Command Prompt**
2. If you have H:\. Enter cd /d C:Users, then you will have C:\Users>
3. Then go to your folder
4. **git clone** [**https://github.com/lingjiekong91/GithubClass.git**](https://github.com/lingjiekong91/GithubClass.git)**.** Then you will have a folder named GithubClass in your original folder. GithubClass is a git folder

* **Create a branch**
  + From local
    - 1. Create a local branch

If you want the new branch has the same contents as master, activate master branch before you create a new branch.

git checkout master

git checkout –b <branch>

* + - 1. Push the local branch to the remote repository

git push –u origin <branch>

* + Download a existing remote branch to your local

git checkout –b <local\_branch\_name> origin/<remote\_branch\_name>

* **Delate a branch**
  + Local

Git branch -D <branch>

* + Remote

Git push origin --delete <branch>

* **Download from Remote master to your local branch**
* Upload your local branch to your remote branch (**After you complete all your changes**)
  + Complete your job, save it.

You can find all your change from GitHub Desktop Application

* + - 1. With “+” means you add this line
      2. With “-“ means you delete this line
      3. Make a commit to your change, so that your partner can understand why you made such changes. You can commit on the left-down corner.
      4. Click Commit to XXX(Your branch name)
      5. Click **Push origin**
  + Open GitHub Page

1. **Pull requests**
2. Compare your branch to master
3. Base: master, compare: Your Branch
4. **Create Pull request**
   * Owner of this repository has the right to confirm your request