Welcome to ECE2800J-25FA

Welcome to ECE2800J-25FA! This page is your personal repository. Throughout the semester, you will submit your exercises and projects here. Below, you'll find detailed information and instructions on how to submit your homework.

1. Basic Setup

You need to do some preparations before submitting your homework to Gitea, testing the correctness through JOJ3 and releasing your homework.

1.1 Git Setup

You can follow the official installation tutorial to install git on different operating systems:

- Git Installation Tutorial for **Windows**
- Git Installation Tutorial for **Mac**
- Git Installation Tutorial for **Linux**

1.2 Gitea SSH Key Setup

First, you need to generate your SSH Key. Open your terminal and run the following command:

```
$ ssh-keygen -t ed25519 -C "<jAccount>@sjtu.edu.cn"
```

NOTE ! `<jAccount>` should be replaced with your actual jAccount.

Then, press the `Enter` key until the prompts end. A file called `id_ed25519.pub` will be created. Print out the file by running the following command:

```
$ cat ~/.ssh/id_ed25519.pub
```

Copy the content of this file and navigate to User settings for SSH/GPG Keys. Click `Add Key` and enter the `Key Name` and `Content` :

- `Key Name` : Anything you like.
- `Content` : The content you just copied from file `id_ed25519.pub` .

1.3 Initial Git Repository Setup

This part should be completed only once. First, run the following command:

```
$ git clone ssh://git@focs.ji.sjtu.edu.cn:2222/ece280s1/<repo>.git
```

NOTE ! `<repo>` should be replaced by the name of your personal repository.

Then, a directory named exactly `<repo>` will be downloaded to your local machine. After that, make the necessary configurations:

```
$ cd <path_to_repo>
$ git config user.name <name>
$ git config user.email <jAccount>@sjtu.edu.cn
```

- `<path_to_repo>` : replaced by the absolute path of directory of your local repository.
- `<name>` : replaced by your name in English.
- `<jAccount>` : replaced with your actual jAccount.

2. Homework Submission Workflow

After completing the setup, you can write and submit your homework to Gitea. The following outlines the workflow for completing a homework assignment.

2.1 Locally Write and Test

The teaching team will upload homework assignments according to the course syllabus. Once a homework assignment is uploaded to Canvas, you should download its description and related files to your local machine. Then, write your code in the local directory and thoroughly test the correctness of your code. You should be confident in your code before proceeding to the next step in the workflow.

2.2 Upload to Remote Repository

Based on your homework's name, copy your code to the corresponding directory in your local repository.

- projects: `p1` or `p2` or `p3` or `p4` or `p5`
- exercises: `ex2` or `ex3` or `ex4` or `ex5` or `ex6`

NOTE ! Always run the following command before you want to change anything in your locol repository:

```
$ git pull
```

NOTE ! Carefully read the `README.md` file in the corresponding directory and avoid copying unnecessary files or code.

Then, open the terminal and upload your code to remote repository by running the following commands:

```
$ cd <path_to_repo>
$ git add <name_of_homework>
$ git commit -m "<commit_message>"
$ git push
```

- `<path_to_repo>` : replaced by the absolute path of directory of your local repository.
- `<name_of_homework>` : replaced by the name of the homework (`p1` or `p2` or `p3` or `p4` or `p5` or `ex2` or `ex3` or `ex4` or `ex5` or `ex6`).
- `<commit_message>` : replaced by properly formatted commit message according to the changes, which will be introduced in **3.1 Commit messages**.

NOTE ! If you want to make an empty commit (where nothing changes compared to your last commit), you can add the `--allow-empty` argument.

```
$ git commit --allow-empty -m "<commit_message>"
```

Finally, check this page to see if your remote repository has been successfully updated.

2.3 Receive Feedback and Fix Errors

Click `Issues` on this page and navigate to the corresponding issue based on your recently submitted homework. The issue title will look something like this:

```
JOJ3 Result for <name_of_homework> by @<jAccount> - Score: <current_score> / 100
```

- `<name_of_homework>` : replaced by the name of the homework (`p1` or `p2` or `p3` or `p4` or `p5` or `ex2` or `ex3` or `ex4` or `ex5` or `ex6`).
- `<jAccount>` : replaced with your actual jAccount.

- `<current_score>` : replaced with the current score of basic cases + code quality of the homework.

NOTE ! You can see detailed information by clicking the small arrow pointing to the right.

Then, return to your code, fix the errors, thoroughly test it locally, and reupload the updated version.

2.4 Release

Release means submitting the final version of your homework and receiving feedback of hidden cases.

NOTE ! You should first make sure the `scope` of your latest commit is exactly the projects (`p1` or `p2` or `p3` or `p4` or `p5`) or exercises (`ex2` or `ex3` or `ex4` or `ex5` or `ex6`) you want to release and submit.

Click `Releases` on this page, then select `New Release` . Enter the `Tag Name` and `Release Title` , ensuring both match the `scope` of your latest commit. Finally, click `Publish Release` to successfully submit your homework.

After release, click `Issues` on this page and navigate to the corresponding issue based on your recently submitted homework to see the score of hidden cases. The issue title will look something like this:

```
JOJ3 Result for <name_of_homework>_h by @<jAccount> - Score: <hidden_cases> / 100
```

- `<name_of_homework>` : replaced by the name of the homework (`p1` or `p2` or `p3` or `p4` or `p5` or `ex2` or `ex3` or `ex4` or `ex5` or `ex6`).
- `<jAccount>` : replaced with your actual jAccount.
- `<hidden_cases>` : replaced with the score of hidden cases of the homework.

The only difference between the titles of the two issues is that the one for hidden cases includes `_h` .

WARNING !!! For each project or exercise, you are only allowed to release once, and you cannot delete any issues. If you release more than once or delete any issue, heavy deductions will follow(In the previous section, more than one release or delete issue result in getting 0 in that single homework).

3. Appendix

3.1 Commit messages

All commit messages are expected to comply with Conventional Commits specifications. JOJ3 parses your commit message and based on its content will trigger different sets of actions. Every commit must be in the format:

```
$ git commit -m "type(scope): message [option_list]"
$ git commit --allow-empty -m "type(scope): message [option_list]"
```

NOTE ! The `type` must be one of:

- `feat` : the commit adds a new feature
- `fix` : the commit fixes a bug
- `docs` : the commit only contains changes related to documentation
- `style` : the commit only adjusts the style, eg. add spaces, tabs, reformat existing code
- `refactor` : changes to the code which neither add a feature nor fix a bug
- `perf` : a commit which helps improve performance
- `test` : adding or correcting tests
- `build` : changes affecting the build system, eg. Makefile, dependencies
- `ci` : commit related to Continuous Integration (CI), eg. Gitea actions configuration
- `chores` : changes not modifying source code or test cases
- `revert` : revert a previous commit

NOTE ! The `scope` must be one of:

- `p1` or `p2` or `p3` or `p4` or `p5` : submit corresponding projects
- `ex2` or `ex3` or `ex4` or `ex5` or `ex6` : submit corresponding exercises

NOTE ! The `message` should concisely describe the changes included in this commit.

NOTE ! The `option_list` can be empty or contain one or both:

- `build` : triggers compilation of your program
- `joj` : tests your program for correctness on JOJ3

For example, the following commits are valid:

```
$ git commit --allow-empty -m "feat(p1): the last submission of p1 [build joj]"
$ git commit -m "fix(p3): memory leak fixed [build joj]"
$ git commit -m "style(ex5): improve code quality [build]"
$ git commit -m "revert(ex4): undo latest code []"
$ git commit -m "docs(ex2): add comments []"
```

3.2 Honor Code

You **MUST NOT** do the following things:

- discuss the solutions to the homework (exercises and projects) with others
- directly copy someone else's work
- use others' testing programs
- use any artificial intelligence (ChatGPT, DeepSeek, Copilot and so on)

The teaching team will **review** every submission to check for **similarities**. Such actions will be considered a violation of the Honor Code!!!