

# COMP1035 – Lab 01: Group Version Control

## Objectives:

1. To collaboratively produce documentation in group's git repository.
  2. To demonstrate understanding of group-level version control.
  3. To effectively manage any merge conflicts encountered.
- 

## Getting Used to Git and Markdown – Using Command Line Interface (CLI)

### Git Task 1

#### Step 1: Forking the Remote Repository (To be completed by a **selected member (e.g., group leader) ONLY**)

- a. Log in to <https://csprojects.nottingham.edu.cn> using your university ID and password.
- b. Under "Projects", create a new project for the group by clicking on "New project" blue button. Choose the "Create blank project".
- c. The project name should be your group's name. Ensure the "Visibility Level" is set to "Private". Ignore other options for now.
- d. Click the "Create project" blue button.
- e. Check the "[Lab01-HowToForkAndClone.pdf](#)" and follow instructions under "Task A" to fork the repository template to your group's location.

**IMPORTANT:** Once the selected member completes the first step, that member can add other group members to the group repository. Search for group member names and add them with the role "Maintainer" (**ensure the group members have log in to the csprojects at least ONCE**).

#### Step 2: Obtaining a Local Copy of the Repository (All Members)

- a. Git clone **YOUR TEAM's** forked repository to your preferred location on your local machine (laptop). **DO NOT CLONE from the "comp1035\_fse/comp1035-template repository"**.
- b. Follow the instructions under "Task B" in "[Lab01-HowToForkAndClone.pdf](#)".
- c. Once cloned, navigate into the repository directory by typing '`cd <team name folder>`' in your terminal or command prompt.
- d. Navigate into the main folder to locate README.md.
- e. Type '`git status`' to confirm you have the latest version of the repository.

### Markdown Task 1

#### Step 3: Creating a Brief Markdown Profile in the "Contributors" Folder (All Members)

- a. Create a `<yourname>.md` (e.g., `bryanlbg.md`) in your "contributors" folder.

- b. Add details about yourself to this created markdown file (see section G1 on the final page for detailed markdown guideline), including at least your name, email address (as a link), and ideally, a photo (place the photo in the images folder).
- c. Additional details such as hobbies, interests, and achievements are optional but encouraged. A sample from last year is provided on the final page for reference. You can edit the file using vim (see Section G2) or nano (see Section G3). Refer to the "[Markdown Cheat Sheet](#)" for formatting styles.

## Git Task 2

### Step 4: Synchronize a File without Conflicts (Ideal Case).

- a. Check '*git status*' to ensure only your edited file is displayed.
- b. Add your changes to the staging area by typing '*git add <your-profile>.md*'.
- c. Commit your changes with a descriptive message using '*git commit -m "<message about adding your profile>"*'. Use a sensible commit message such as "Documentation: Updated Bryan's profile in the 'contributors' folder." Refer to "[Writing Good Commit Messages](#)" for guidelines (Expected Readings of Week 1 on Moodle).
- d. Push your changes to the remote repository with '*git push*'.
- e. As each of you edits different files, this should proceed smoothly. If not, try '*git pull*' first to merge any changes from the server with your local repository before pushing your changes.

## Markdown Task 2

### Step 5: Adding a Link to Your Markdown Profile in the Main README.md

- a. In the provided README.md starter file, there is a space allocated for a list of contributors. Add your name as a link in the list, referencing the link "*contributors/<yourname>.md*" rather than a full [https://](#) style link.
- b. This may lead to conflicts as multiple members edit the same file, which will be addressed in the next step.

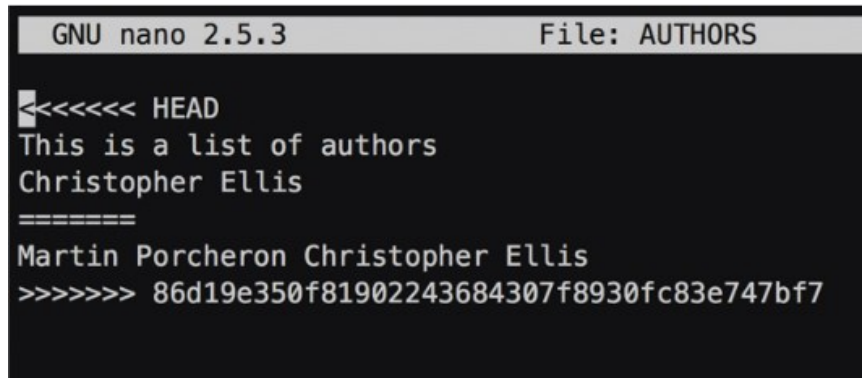
## Git Task 3

### Step 6: Synchronizing a File with Conflicts

- a. Repeat *Git Task 2* for the main README.md.
- b. If you encounter merge conflict warnings, fetch the latest version of the report with '*git pull*'. This will attempt to merge changes, and any further merge errors can be viewed using '*git status*'.
- c. Resolve conflicts by editing the file to retain desired changes.
- d. After resolving conflicts, push your changes to the remote repository again.

**Hint:** Handling merge conflicts can be challenging, so refer to lecture materials or online resources for assistance.

If using nano, a merge conflict will appear as follows:



```
GNU nano 2.5.3 File: AUTHORS
<<<<<< HEAD
This is a list of authors
Christopher Ellis
=====
Martin Porcheron Christopher Ellis
>>>>>> 86d19e350f81902243684307f8930fc83e747bf7
```

The lines between <<<<<< HEAD and ===== representing the local changes, and ===== to >>>>>> representing external changes by someone else.

You need to delete the <<<<<<, ===== and >>>>>> lines and merge the remaining lines as desired. Then exit nano/vim to complete the merge/push.

### Finishing Activity

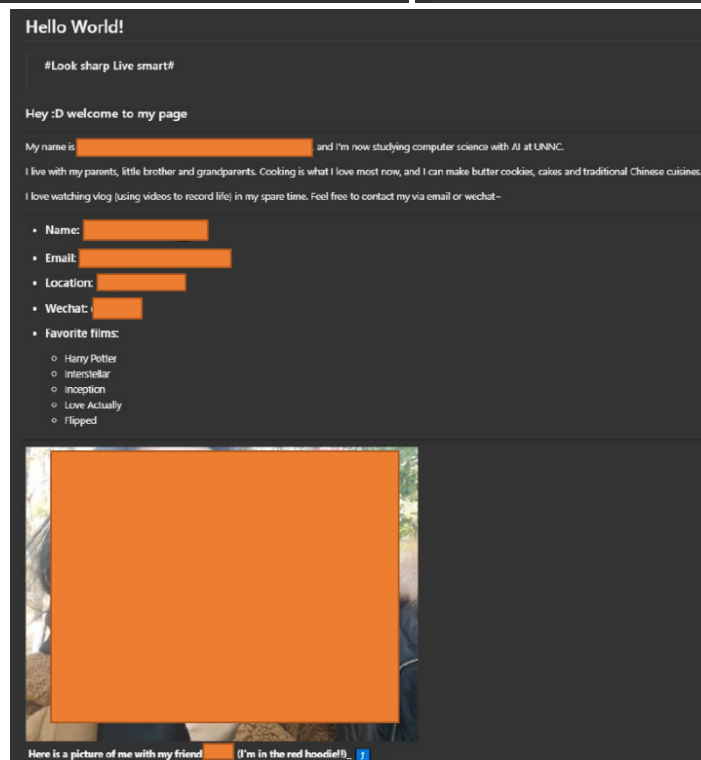
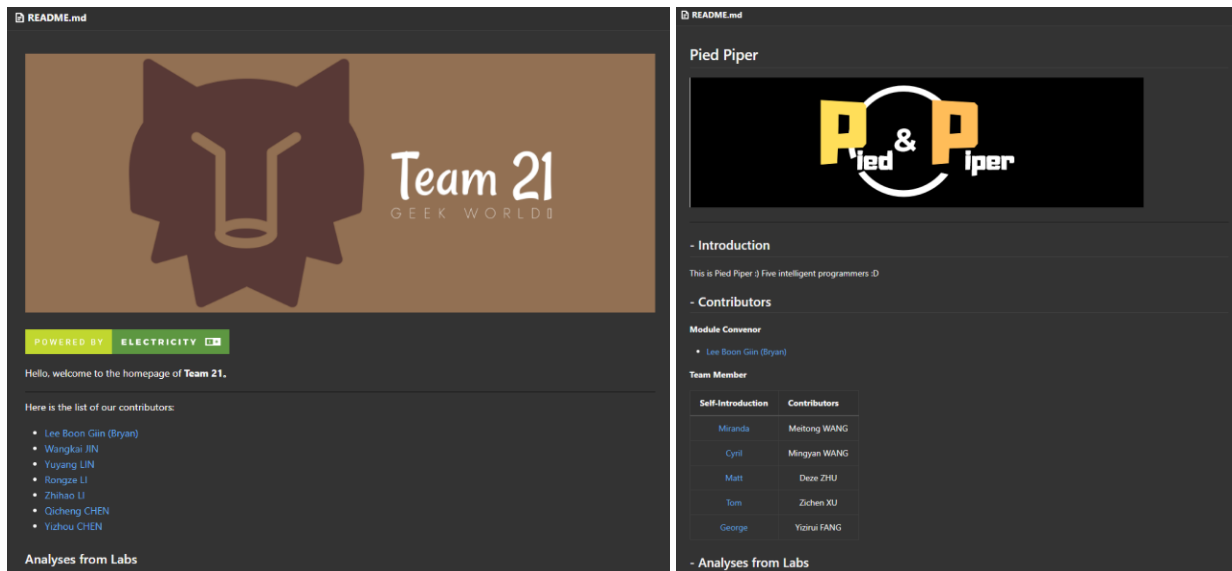
Once all members complete the above steps, perform '*git pull*' to ensure everyone has the final version.

---

## Group Activity – Setup Repository for Your Group

This activity should be completed as a group.

1. Edit the main README.md page as desired to provide information about your group. Examples from previous years are provided below.
2. Create milestones, labels, and issues/issue boards for your group.
3. Practice creating, labelling, and closing issues.
4. Use the main README.md to link to work done throughout the semester, including lab work and coursework submissions. Consider creating headings for these sections.



Good Samples of Team and Contributor Profile from Previous Years.

## Guides List

- G1. [GitLab Markdown Documentation](#)
- G2. [Vim for Beginner Tutorial](#)
- G3. [The Beginner's Guide to Nano Editor](#)
- G4. [Git Installation](#)
- G5. [Setting SSH Key in GitLab](#)