

Lab Sheet 1: Version Control with Git & GitHub

Objectives

By completing today's workshop, you will:

- Configure **Git** on your local machine.
 - Create a **GitHub Account** and link it to your computer.
 - Understand the basic Git workflow: **Add**, **Commit**, and **Push**.
 - Learn how to **Fork** and contribute to open source projects.
-

Task 1: First-Time Setup & The Local Repo

Goal: Configure your identity and create your first local commit.

1. **Open the Terminal:** Open Git Bash (Windows) or Terminal (Mac/Linux).
2. **Configure Identity:** (Run these one by one)

```
git config --global user.name "Your Name"  
git config --global user.email "your.email@lincoln.ac.uk"
```

Note that the email should match the one you will use for GitHub (we suggest your uni email). your name can be anything you like.

3. **Create a Project:** This will be your first local repository. You can create it anywhere on your computer, and you can use this as the repository for your final CMP9134 project:

```
mkdir CMP9134_final_project  
cd CMP9134_final_project  
git init
```

You should see a message saying "Initialized empty Git repository".

4. **Create & Commit:**

- Create a file named `hello.txt` and add some text to it. On a windows terminal, you can use: `echo "" > hello.txt` and then edit it with Notepad. You can also create and edit the file with any editor you like.
- Check status: `git status` (Should be red).
- Stage it: `git add hello.txt` (Check status: Should be green).
- Commit it: `git commit -m "My first commit"`

5. **Verify:** Run `git log` to see your commit history. Note the commit ID, author, date, and message.

Task 2: Create a GitHub Account

Goal: Set up your cloud identity.

- Go to github.com/join.
- Create an account using your **University Email Address** (recommended for academic benefits) or use your existing personal account if you already have one.
- Verify your email address to ensure you can create repositories.

Task 3: Connecting to the Cloud

Goal: Upload your local code to GitHub.

1. **Create Remote Repo:** Click the + icon → **New Repository** on GitHub.

- Name: `CMP9134_final_project` (or any name you like).
- Visibility: **Public** (or Private – note that your final project repository should be public at the time of submission at the end of the semester!).
- **Do NOT** check "Add README" or ".gitignore" (keep it empty).

2. **Link & Push:** Copy the URL provided by GitHub (ends in `.git`) and run:

```
git remote add origin <PASTE_YOUR_URL_HERE>
git branch -M main
git push -u origin main
```

3. **Verify:** Refresh your GitHub page. You should see your `hello.txt` file!

Task 4: The Full Workflow

Goal: Practice the cycle of modifying and updating code.

1. **Modify:** Open `hello.txt` and add a second line of text.
2. **Add Code:** Create a new file `MyCode.py` which prints a basic "Hello World" in Python.
3. **Check Status:** Run `git status`.
4. **Stage All:** Run `git add .` to stage both changes.
5. **Commit:** Run `git commit -m "Added Python program and updated text"`.
6. **Push:** Run `git push`.

Task 5: Forking a Repository & Pull Requests

Goal: Learn how to contribute to existing projects.

What is Forking? Forking creates a personal copy of someone else's project. It allows you to freely experiment with changes without affecting the original project. This is the standard way to contribute to Open Source software.

What is a Pull Request (PR)? When you modify your fork, the original repository doesn't change. To send your changes back, you must open a **Pull Request**.

- Think of it like a journalist submitting an article to an editor.
- You are requesting that the owner “**pulls**” your changes into their project.
- This allows for code review and discussion before the changes are accepted.

Task Steps:

1. **Find the Repo:** Go to <https://github.com/octocat/Spoon-Knife>. This is a standard test repo provided by GitHub.
2. **Fork It:** Click the ”**Fork**” button in the top-right corner. This creates a copy under *your* account.
3. **Clone Your Fork:**
 - On your version of the page, click **Code** and copy the URL.
 - In your terminal: `git clone <YOUR_FORK_URL>`
4. **Modify & Push:**
 - Create a file named `Week13_<YourName>.txt` inside the folder.
 - Add, Commit, and Push.
 - Go to GitHub and see that the change exists on *your* fork, but not the original `octocat` repository.
5. **Create a Pull Request:**
 - Go to your fork on GitHub.
 - Click the **Compare & pull request** button, or go to the **Pull requests** tab and click **New pull request**.
 - Add a title and description, then click **Create pull request**.
 - This notifies the original repository owner of your proposed changes. They can review and choose to merge them.
 - **Note:** Pull Requests can be created also to merge changes from a branch into the main branch of your own repository. It provides a way for the team to review changes before integrating them.

Further Resources

- **Visualizing Git:** <https://git-school.github.io/visualizing-git/> (Great for understanding branching).
- **Oh My Git!** <https://ohmygit.org/> (An open source game to learn Git).
- **GitHub Docs (Forking):** <https://docs.github.com/en/get-started/quickstart/fork-a-repo>
- **GitHub Docs (Pull Requests):** <https://docs.github.com/en/pull-requests/collaborating-with-pull-requests/proposing-changes-to-your-work-with-pull-requests/about-pull-requests>
- **Git Cheatsheet:** <https://education.github.com/git-cheat-sheet-education.pdf>