

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
KHOA AN TOÀN THÔNG TIN**



BÁO CÁO TUẦN 1 THỰC TẬP TỐT NGHIỆP

Tìm hiểu về Git

Sinh viên thực hiện:

B21DCAT090 Nguyễn Minh Hiệu

Giảng viên hướng dẫn: TS Nguyễn Ngọc Điệp

HÀ NỘI 6-2025

Mục Lục

I. Git là gì? Vì sao cần dùng Git?	3
II. Các khái niệm cơ bản trong Git	3
III. Quy trình làm việc cơ bản với Git	3
1. Khởi tạo kho Git:	3
2. Kết nối với remote repository (ví dụ GitHub):	4
3. Làm việc với nhánh (branch)	4
4. Một số lệnh Git hữu ích khác	4
5. Quy trình làm việc nhóm chuyên nghiệp với Git	4
IV. Bài tập thực hành: Tạo và quản lý dự án với Git	5
1. Chuẩn bị	5
2. Khởi tạo dự án mới	5
3. Tạo file và commit đầu tiên	6
4. Kết nối với GitHub và đẩy (push) lên remote	7
5. Làm việc với nhánh (branch)	8
6. Gộp nhánh (merge) và xử lý xung đột (nếu có)	9
7. Xóa nhánh đã hoàn thành (tùy chọn)	10

I. Git là gì? Vì sao cần dùng Git?

Git là hệ thống quản lý phiên bản phân tán (Distributed Version Control System – DVCS) phổ biến nhất hiện nay, giúp bạn:

- **Theo dõi lịch sử thay đổi** của mã nguồn, tài liệu, v.v.
- **Làm việc nhóm** hiệu quả, tránh ghi đè/chồng chéo mã.
- **Quản lý các phiên bản**, nhánh phát triển, kiểm soát lỗi và phục hồi nhanh chóng.

Git lưu trữ dữ liệu dưới dạng **snapshot** (ảnh chụp trạng thái toàn bộ dự án tại mỗi lần commit), thay vì chỉ lưu các thay đổi từng dòng như nhiều VCS khác. Điều này giúp Git nhanh, bảo toàn dữ liệu, dễ kiểm soát lịch sử và cực kỳ linh hoạt trong làm việc nhóm.

II. Các khái niệm cơ bản trong Git

Khái niệm	Ý nghĩa
Repository	Kho lưu trữ dự án, chứa toàn bộ mã nguồn và lịch sử thay đổi.
Commit	Một lần lưu lại trạng thái dự án (snapshot), kèm mô tả thay đổi.
Branch	Nhánh phát triển song song, giúp thử nghiệm/chia nhỏ công việc mà không ảnh hưởng main.
Merge	Gộp thay đổi từ nhánh này sang nhánh khác.
Clone	Tạo bản sao của repository về máy cá nhân.
Remote	Repository trên máy chủ (GitHub, GitLab, Bitbucket...), dùng để chia sẻ, đồng bộ nhóm.
Staging Area	Vùng tạm, nơi bạn chọn các thay đổi sẽ commit.
Working Directory	Thư mục làm việc thực tế trên máy bạn.

III. Quy trình làm việc cơ bản với Git

1. Khởi tạo kho Git:

`git init`

Cấu hình thông tin cá nhân (làm một lần):

`git config --global user.name "Tên của bạn"`

`git config --global user.email "email@domain.com"`

Thêm file vào vùng staging:

`git add <tên_file>`

`git add .`

Lưu thay đổi (commit):

git commit -m "Mô tả ngắn gọn về thay đổi"

2. Kết nối với remote repository (ví dụ GitHub):

git remote add origin <url-repo-trên-github>

Đẩy (push) thay đổi lên remote:

git push -u origin main *# Lần đầu, hoặc*

git push *# Các lần sau*

Lấy (pull) thay đổi mới nhất từ remote:

git pull

3. Làm việc với nhánh (branch)

- **Tạo nhánh mới:**

git branch <ten_nhanh>

- **Chuyển sang nhánh khác:**

git checkout <ten_nhanh>

- **Tạo và chuyển nhánh cùng lúc:**

git checkout -b <ten_nhanh>

- **Gộp nhánh** (ví dụ gộp nhánh feature vào main):

git checkout main

git merge feature

- **Xem danh sách nhánh:**

git branch

4. Một số lệnh Git hữu ích khác

- **Kiểm tra trạng thái file:**

git status

- **Xem lịch sử commit:**

git log

- **Xem ai sửa dòng nào trong file:**

git blame <ten_file>

- **Hoàn tác thay đổi chưa commit:**

git checkout -- <ten_file>

- **Hoàn tác commit gần nhất:**

git reset --soft HEAD~1

5. Quy trình làm việc nhóm chuyên nghiệp với Git

- **Clone repository về máy cá nhân:**

git clone <url-repo>

- **Tạo nhánh riêng cho từng tính năng/bugfix.**
- **Commit, push nhánh lên remote.**
- **Tạo pull request (PR)** trên GitHub/GitLab để xin review & merge vào main.
- **Giải quyết xung đột (conflict)** nếu có, sử dụng git merge hoặc git rebase tùy workflow.

IV. Bài tập thực hành: Tạo và quản lý dự án với Git

Đăng xuất tài khoản GitHub cũ nếu tài khoản bị khóa

1. Mở **Control Panel** → **User Accounts** → **Credential Manager**.
2. Chọn **Windows Credentials**.
3. Tìm các mục có chứa git: hoặc github.com trong tên.
4. Nhấn vào từng mục và chọn **Remove** để xóa thông tin đăng nhập cũ

1. Chuẩn bị

- Cài đặt **Git** trên máy tính (tải từ git-scm.com).
- Đăng ký tài khoản **GitHub** (hoặc GitLab/Bitbucket).

2. Khởi tạo dự án mới

Bước 1: Tạo một thư mục mới cho dự án trên máy tính.

```
mkdir demo-git
```

```
cd demo-git
```

Bước 2: Khởi tạo Git repository.

```
git init
```

Bước 3: Cấu hình thông tin cá nhân (chỉ cần làm một lần).

```
git config --global user.name "Tên của bạn"
```

```
git config --global user.email email@domain.com
```

```
MINGW64:/c/Users/Dell/demo-git
Dell@DESKTOP-7N5AC72 MINGW64 ~ (main)
$ mkdir demo-git

Dell@DESKTOP-7N5AC72 MINGW64 ~ (main)
$ cd demo-git/

Dell@DESKTOP-7N5AC72 MINGW64 ~/demo-git (main)
$ git init
Initialized empty Git repository in C:/Users/Dell/demo-git/.git/

Dell@DESKTOP-7N5AC72 MINGW64 ~/demo-git (master)
$ git config --global user.name [redacted]

Dell@DESKTOP-7N5AC72 MINGW64 ~/demo-git (master)
$ git config --global user.email [redacted]

Dell@DESKTOP-7N5AC72 MINGW64 ~/demo-git (master)
$ .....
```

3. Tạo file và commit đầu tiên

Bước 4: Tạo file mới, ví dụ: README.md

```
echo "# Demo Git" > README.md
```

Bước 5: Kiểm tra trạng thái file.

```
git status
```

Bước 6: Thêm file vào vùng staging.

```
git add README.md
```

Bước 7: Lưu lại thay đổi (commit).

```
git commit -m "Thêm file README.md đầu tiên"
```

```
MINGW64:/c/Users/Dell/demo-git
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        README.md

nothing added to commit but untracked files present (use "git add" to track)

Dell@DESKTOP-7N5AC72 MINGW64 ~/demo-git (master)
$ git add README.md
warning: in the working copy of 'README.md', LF will be replaced by CRLF the next time Git touches it

Dell@DESKTOP-7N5AC72 MINGW64 ~/demo-git (master)
$ git commit -m "Thêm file README.md đầu tiên"
[master (root-commit) 8be069a] Thêm file README.md đầu tiên
1 file changed, 1 insertion(+)
create mode 100644 README.md

Dell@DESKTOP-7N5AC72 MINGW64 ~/demo-git (master)
$
```

4. Kết nối với GitHub và đẩy (push) lên remote

Bước 8:

- Tạo một repository mới trên GitHub (không chọn "Initialize with README").
- Copy URL của repository (ví dụ: <https://github.com/tenuser/demo-git.git>).

Bước 9: Kết nối repository local với remote.

```
git remote add origin https://github.com/tenuser/demo-git.git
```

Bước 10: Đẩy commit đầu tiên lên GitHub.

```
git push -u origin main
```

Nếu báo lỗi về nhánh main chưa tồn tại, dùng:

```
git branch -M main
```

```
git push -u origin main
```

```
MINGW64:/c/Users/Dell/demo-git
Dell@DESKTOP-7N5AC72 MINGW64 ~/demo-git (main)
$ echo "# demo-git" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/hieunm2025/demo-git.git
git push -u origin main
Reinitialized existing Git repository in C:/Users/Dell/demo-git/.git/
warning: in the working copy of 'README.md', LF will be replaced by CRLF the next time Git touches it
[main 2266582] first commit
1 file changed, 1 insertion(+)
info: please complete authentication in your browser...
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (9/9), 720 bytes | 720.00 KiB/s, done.
Total 9 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/hieunm2025/demo-git.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

5. Làm việc với nhánh (branch)

Bước 11: Tạo nhánh mới để phát triển tính năng.

```
git checkout -b feature-hello
```

```
MINGW64:/c/Users/Dell/demo-git
Dell@DESKTOP-7N5AC72 MINGW64 ~/demo-git (main)
$ git checkout -b feature-hello
Switched to a new branch 'feature-hello'

Dell@DESKTOP-7N5AC72 MINGW64 ~/demo-git (feature-hello)
$
```

Bước 12: Thêm file mới, ví dụ: hello.txt

```
echo "Xin chào Git!" > hello.txt
```

```
git add hello.txt
```

```
git commit -m "Thêm file hello.txt"
```



```
MINGW64:/c/Users/Dell/demo-git

Dell@DESKTOP-7N5AC72 MINGW64 ~/demo-git (main)
$ git checkout -b feature-hello
Switched to a new branch 'feature-hello'

Dell@DESKTOP-7N5AC72 MINGW64 ~/demo-git (feature-hello)
$ echo "Xin chào Git!" > hello.txt

Dell@DESKTOP-7N5AC72 MINGW64 ~/demo-git (feature-hello)
$ git add hello.txt
warning: in the working copy of 'hello.txt', LF will be replaced by CRLF the next time Git touches it

Dell@DESKTOP-7N5AC72 MINGW64 ~/demo-git (feature-hello)
$ git commit -m "Thêm file hello.txt"
[feature-hello 125c8a0] Thêm file hello.txt
1 file changed, 1 insertion(+)
create mode 100644 hello.txt

Dell@DESKTOP-7N5AC72 MINGW64 ~/demo-git (feature-hello)
$
```

Bước 13: Đẩy nhánh mới lên GitHub.

git push -u origin feature-hello

```
MINGW64:/c/Users/Dell/demo-git

Dell@DESKTOP-7N5AC72 MINGW64 ~/demo-git (feature-hello)
$ git commit -m "Thêm file hello.txt"
[feature-hello 125c8a0] Thêm file hello.txt
1 file changed, 1 insertion(+)
create mode 100644 hello.txt

Dell@DESKTOP-7N5AC72 MINGW64 ~/demo-git (feature-hello)
$ git push -u origin feature-hello
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 305 bytes | 305.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'feature-hello' on GitHub by visiting:
remote:   https://github.com/hieunm2025/demo-git/pull/new/feature-hello
remote:
To https://github.com/hieunm2025/demo-git.git
 * [new branch]      feature-hello -> feature-hello
branch 'feature-hello' set up to track 'origin/feature-hello'.

Dell@DESKTOP-7N5AC72 MINGW64 ~/demo-git (feature-hello)
$
```

6. Gộp nhánh (merge) và xử lý xung đột (nếu có)

Bước 14: Quay lại nhánh main.

git checkout main

Bước 15: Gộp nhánh feature-hello vào main.

git merge feature-hello

```
MINGW64:/c/Users/Dell/demo-git

Dell@DESKTOP-7N5AC72 MINGW64 ~/demo-git (feature-hello)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

Dell@DESKTOP-7N5AC72 MINGW64 ~/demo-git (main)
$ git merge feature-hello
Updating 2266582..125c8a0
Fast-forward
 hello.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 hello.txt

Dell@DESKTOP-7N5AC72 MINGW64 ~/demo-git (main)
$
```

Bước 16: Đẩy thay đổi lên GitHub.

git push

```
MINGW64:/c/Users/Dell/demo-git

Dell@DESKTOP-7N5AC72 MINGW64 ~/demo-git (feature-hello)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

Dell@DESKTOP-7N5AC72 MINGW64 ~/demo-git (main)
$ git merge feature-hello
Updating 2266582..125c8a0
Fast-forward
 hello.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 hello.txt

Dell@DESKTOP-7N5AC72 MINGW64 ~/demo-git (main)
$ git push
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/hieunm2025/demo-git.git
 2266582..125c8a0  main -> main

Dell@DESKTOP-7N5AC72 MINGW64 ~/demo-git (main)
$
```

7. Xóa nhánh đã hoàn thành (tùy chọn)

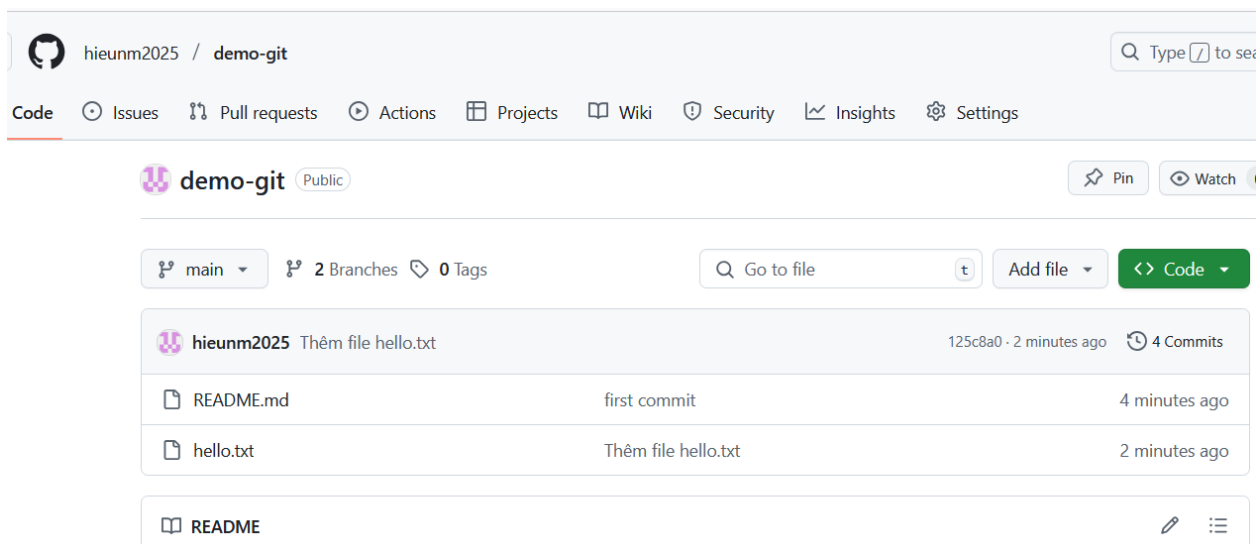
git branch -d feature-hello

git push origin --delete feature-hello

```
MINGW64:/c/Users/Dell/demo-git

Dell@DESKTOP-7N5AC72 MINGW64 ~/demo-git (main)
$ git branch -d feature-hello
Deleted branch feature-hello (was 125c8a0).

Dell@DESKTOP-7N5AC72 MINGW64 ~/demo-git (main)
$
```



Đẩy repo có sẵn lên git

```

MINGW64:/d/Java/BaiTap/src/codeptit
Dell@DESKTOP-7N5AC72 MINGW64 /d/Java/BaiTap/src/codeptit
$ git init
Initialized empty Git repository in D:/Java/BaiTap/src/codeptit/.git/

Dell@DESKTOP-7N5AC72 MINGW64 /d/Java/BaiTap/src/codeptit (master)
$ git remote add origin https://github.com/hieunm2025/Java-OOP.git

Dell@DESKTOP-7N5AC72 MINGW64 /d/Java/BaiTap/src/codeptit (master)
$ git add .

Dell@DESKTOP-7N5AC72 MINGW64 /d/Java/BaiTap/src/codeptit (master)
$ git commit -m "Initial commit"
[master (root-commit) c7998c7] Initial commit
27 files changed, 795 insertions(+)
create mode 100644 BauCu.java
create mode 100644 J01001_HinhChuNhat.java
create mode 100644 J01002_TongNSoNguyenDuongDauTien.java
create mode 100644 J01003_GiaiPhuongTrinhBacNhat.java
create mode 100644 J01004_SoNguyenTo.java
create mode 100644 J01005_ChiaTamGiac.java
create mode 100644 J01006_TinhSoFibonacci.java
create mode 100644 J01007_KiemTraSoFibonacci.java
create mode 100644 J01008_PhanTichThuaSoNguyenTo.java
create mode 100644 J01009_TongGiaiThua.java

Dell@DESKTOP-7N5AC72 MINGW64 /d/Java/BaiTap/src/codeptit (master)
$ git branch -M main

Dell@DESKTOP-7N5AC72 MINGW64 /d/Java/BaiTap/src/codeptit (main)
$ git push -u origin main
Enumerating objects: 32, done.
Counting objects: 100% (32/32), done.
Delta compression using up to 8 threads
Compressing objects: 100% (32/32), done.
Writing objects: 100% (32/32), 11.04 KiB | 1.84 MiB/s, done.
Total 32 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), done.
To https://github.com/hieunm2025/Java-OOP.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

Dell@DESKTOP-7N5AC72 MINGW64 /d/Java/BaiTap/src/codeptit (main)
$

```

Thay đổi file và commit lên git

```
MINGW64:/d/Java/BaiTap/src/codeptit
Dell@DESKTOP-7N5AC72 MINGW64 /d/Java/BaiTap/src/codeptit (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   J04005_KhaiBaoLopThiSinh.java

no changes added to commit (use "git add" and/or "git commit -a")

Dell@DESKTOP-7N5AC72 MINGW64 /d/Java/BaiTap/src/codeptit (main)
$ git add .

Dell@DESKTOP-7N5AC72 MINGW64 /d/Java/BaiTap/src/codeptit (main)
$ git commit -m "Fix J04005: Khai báo lớp thi sinh"
[main c57286f] Fix J04005: Khai báo lớp thi sinh
1 file changed, 24 insertions(+), 5 deletions(-)

Dell@DESKTOP-7N5AC72 MINGW64 /d/Java/BaiTap/src/codeptit (main)
$ git push
Enumerating objects: 5, done.
```