Lab

Basic Git

Objective

- Student understand how Git works.
- Student can use Github for collaborate with teammate.
- Student know basic Git commands.

1. What is Git?

Git is a DevOps tool used for source code management. This is a free and open-source version control system used to handle small to very large projects efficiently. Git is used to tracking changes in the source code, enabling multiple developers to work together on non-linear development [1]

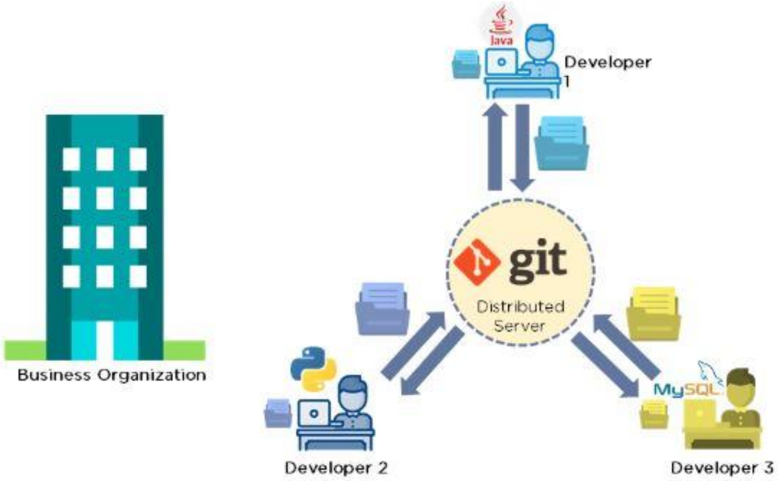


Figure 1 How Git works in Business Organization

ref: https://www.simplilearn.com/tutorials/git-tutorial/what-is-git

2. What is Github

Github is one of popular online software development platform for storing, tracking, and collaborating on software projects. It makes it easy for developers to share code files and collaborate with fellow developers on open-source projects. GitHub also serves as a social networking site where developers can openly network, collaborate, and pitch their work. [2]

Figure 1 Github

ref: https://github.com/

3. Different between Git and Github

Git	Github
Distributed version control system which tracks changes to source code over time	Web-based hosting service for Git repository to bring teams together
Command-line tool that requires an interface to interact with the world	Graphical interface and a development platform created for millions of developers
Creates a local repository to track changes locally rather than store them on a centralized server.	Open-source while means code is stored in a centralized server and is accessible to everybody
Stores and catalogs changes in code in a repository	Provides a platform as a collaborative effort to bring teams together

4. Basic Git command

Command	Meaning
git config –global user.name “name”	Set up username
git config –global user.email “email”	Set up email
git init	Start a new Git repository and start tracking the folder or project you are working on. Along with creating a local repository inside.
git clone <URL>	Copy project’s code from server into own computer
git add <file>	Command for adding files or folders to our project into the Staging Area to tell Git that we want to compile changes to this file in the next commit. “git add .” will add new files and all changed files into the Staging Area.
git commit	Commands are like notes that tell us the changes we've made to the code. The Commit Message should be a clear description of what has been changed.
git status	Commands that show the status of changes as untracked, modified, or staged indicate what we've added, committed, or done to our code.
git push <remote> <branch>	Command used to upload all the code of our local branch that has been committed and added to the Remote Repo.
git pull <remote> <branch>	The commands used to update our local code are the same as the code in the Remote Repo. We can use this command to sync our local repository with the Remote to keep the code up-to-date.
git checkout <branch-name>	Command used to switch to another branch in Working Directory
git checkout -b <branch-name>	Command used to create new branch
git merge <branch>	Command used to merge branch to main branch in Remote Repo
git commit -m “comment”	Command used to confirm changes to files that have been added to a stage. You can add comments by adding -m after them.
git fetch	Command used to check if local and remote files are different or not.

References

[1] https://git-scm.com/

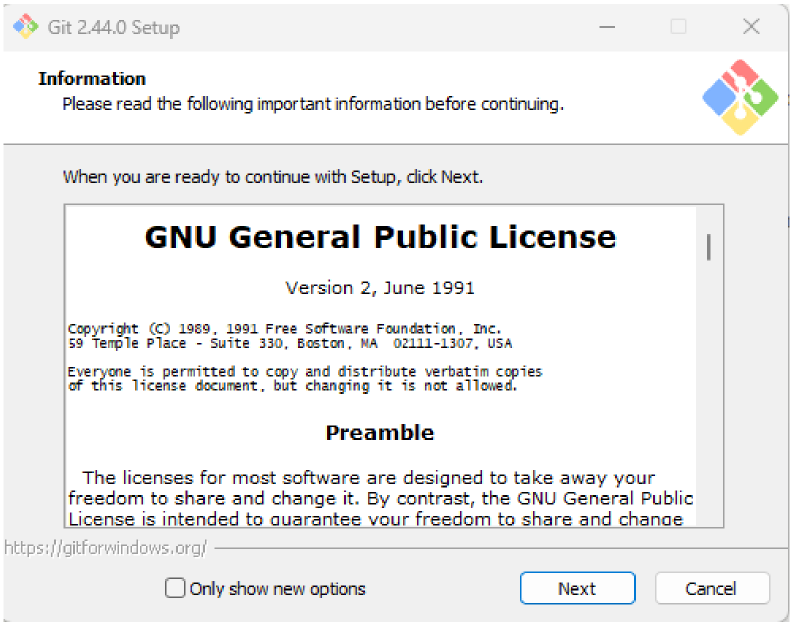
[2] https://github.com/

Part 1: Create download and install Git

1. Go to website https://git-scm.com/ and download Git installation
2. Install Git to your computer

Git install steps

1. Double click Git installation file, the window will show as follow. Then click next



Information

Please read the following important information before continuing.

When you are ready to continue with Setup, click Next.

GNU General Public License

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place - Suite 330, Boston, MA 02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

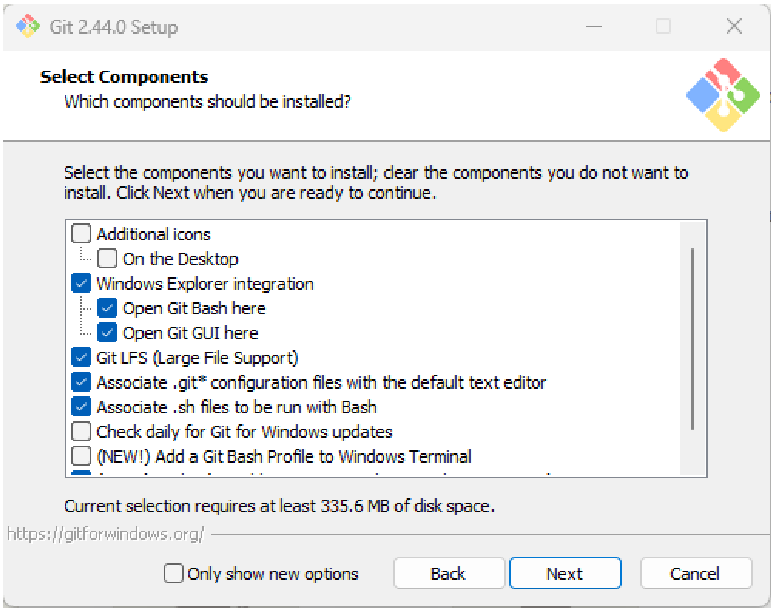
Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change

https://gitforwindows.org/

□ Only show new options

2. Set components as show in the figure, then click next



Select Components

Which components should be installed?

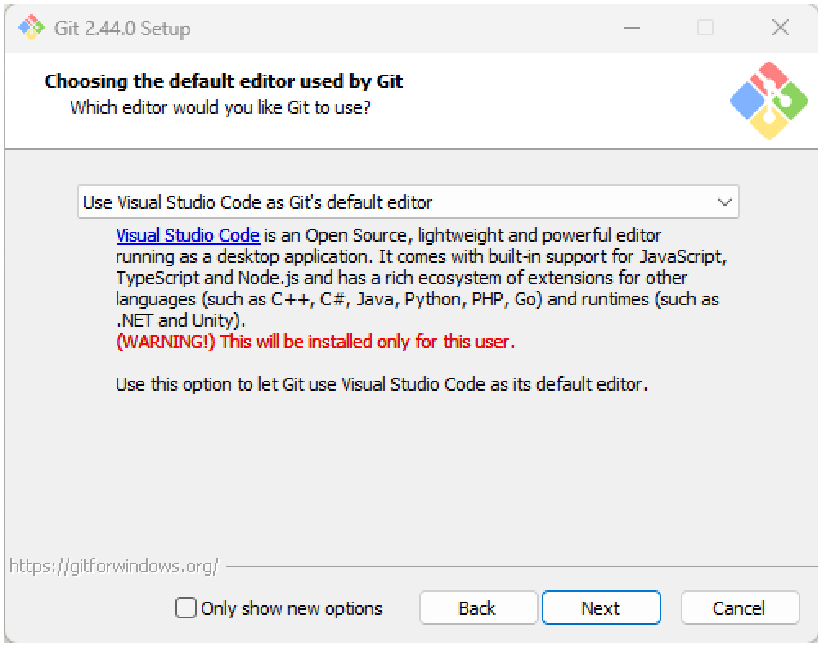
Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

- □ Additional icons
  - □ On the Desktop
- ☑ Windows Explorer integration
  - ☑ Open Git Bash here
  - ☑ Open Git GUI here
- ☑ Git LFS (Large File Support)
- ☑ Associate .git* configuration files with the default text editor
- ☑ Associate .sh files to be run with Bash
- □ Check daily for Git for Windows updates
- □ (NEW!) Add a Git Bash Profile to Windows Terminal

Current selection requires at least 335.6 MB of disk space.

□ Only show new options

3. Choosing the default editor used by Git (Use Visual Studio Code), then click next



Choosing the default editor used by Git

Which editor would you like Git to use?

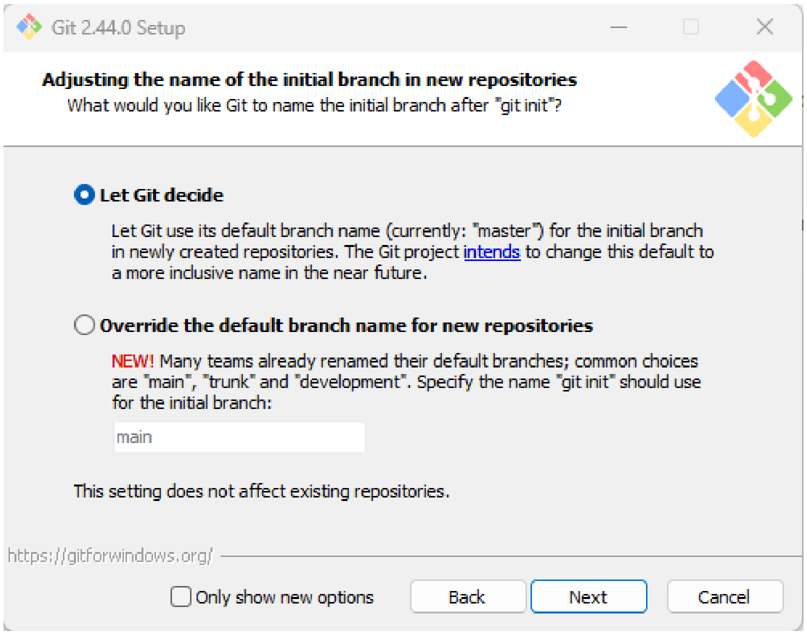
Use Visual Studio Code as Git's default editor

Visual Studio Code is an Open Source, lightweight and powerful editor running as a desktop application. It comes with built-in support for JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions for other languages (such as C++, C#, Java, Python, PHP, Go) and runtimes (such as .NET and Unity).
(WARNING!) This will be installed only for this user.

Use this option to let Git use Visual Studio Code as its default editor.

□ Only show new options

4. Select Let Git decide, then click next



Adjusting the name of the initial branch in new repositories

What would you like Git to name the initial branch after "git init"?

● **Let Git decide**

Let Git use its default branch name (currently: "master") for the initial branch in newly created repositories. The Git project intends to change this default to a more inclusive name in the near future.

○ **Override the default branch name for new repositories**

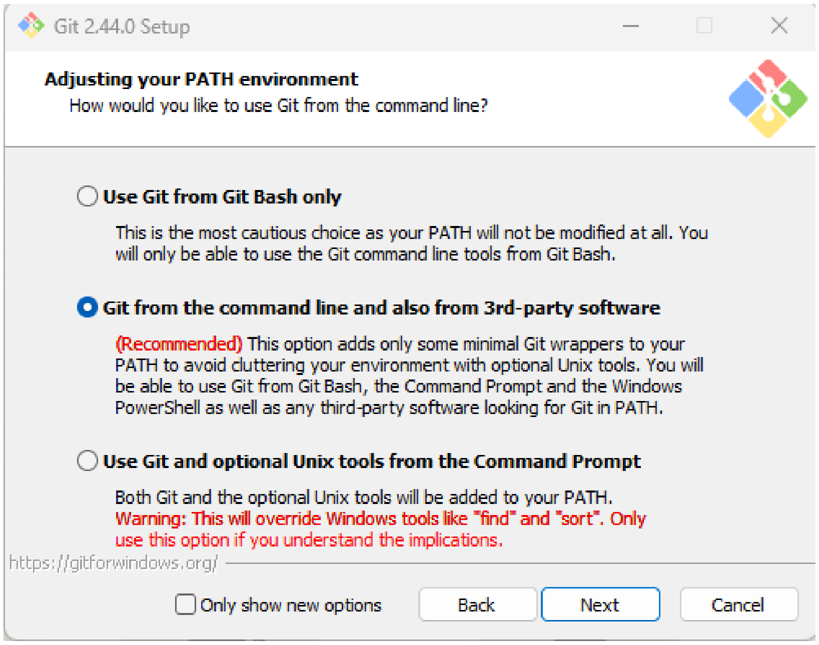
NEW! Many teams already renamed their default branches; common choices are "main", "trunk" and "development". Specify the name "git init" should use for the initial branch:

main

This setting does not affect existing repositories.

https://gitforwindows.org/

5. Select Git from the command line and also from 3rd-party software, then click next



Adjusting your PATH environment

How would you like to use Git from the command line?

○ **Use Git from Git Bash only**

This is the most cautious choice as your PATH will not be modified at all. You will only be able to use the Git command line tools from Git Bash.

● **Git from the command line and also from 3rd-party software**

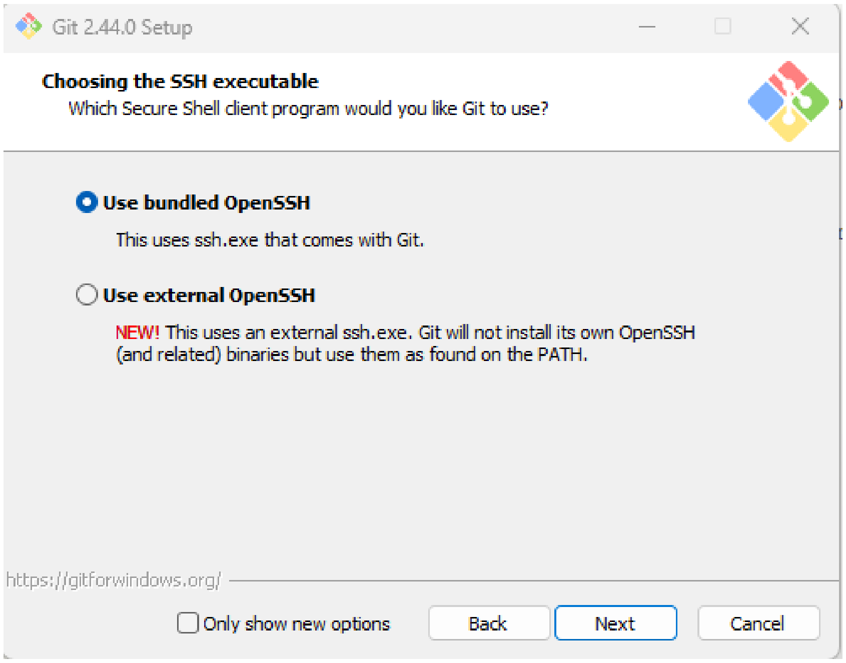
(Recommended) This option adds only some minimal Git wrappers to your PATH to avoid cluttering your environment with optional Unix tools. You will be able to use Git from Git Bash, the Command Prompt and the Windows PowerShell as well as any third-party software looking for Git in PATH.

○ **Use Git and optional Unix tools from the Command Prompt**

Both Git and the optional Unix tools will be added to your PATH. Warning: This will override Windows tools like "find" and "sort". Only use this option if you understand the implications.

https://gitforwindows.org/

6. Select Use bundle OpenSSH, then click next



Git 2.44.0 Setup

Choosing the SSH executable

Which Secure Shell client program would you like Git to use?

● **Use bundled OpenSSH**

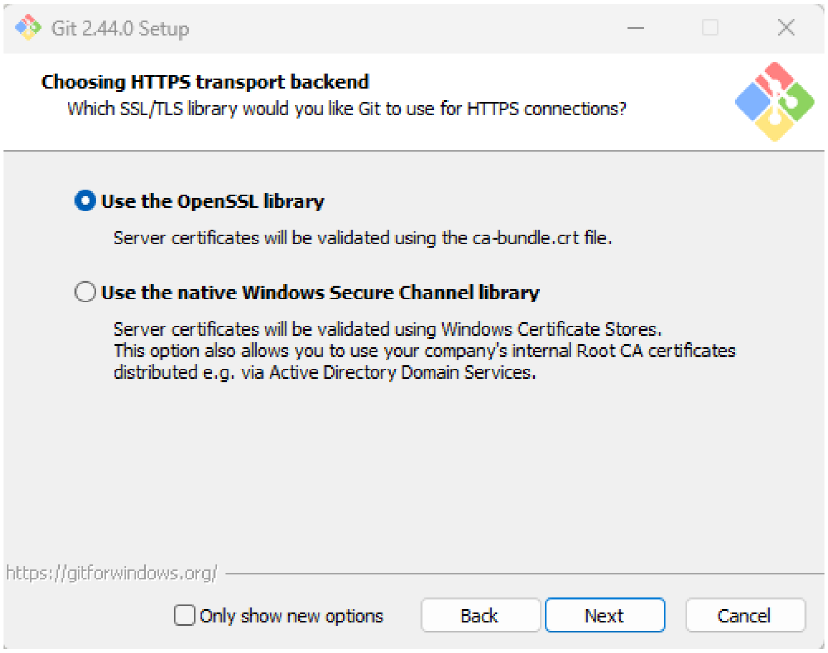
This uses ssh.exe that comes with Git.

○ **Use external OpenSSH**

NEW! This uses an external ssh.exe. Git will not install its own OpenSSH (and related) binaries but use them as found on the PATH.

https://gitforwindows.org/

7. Select Use the OpenSSH library, then click next



Git 2.44.0 Setup

Choosing HTTPS transport backend

Which SSL/TLS library would you like Git to use for HTTPS connections?

● **Use the OpenSSL library**

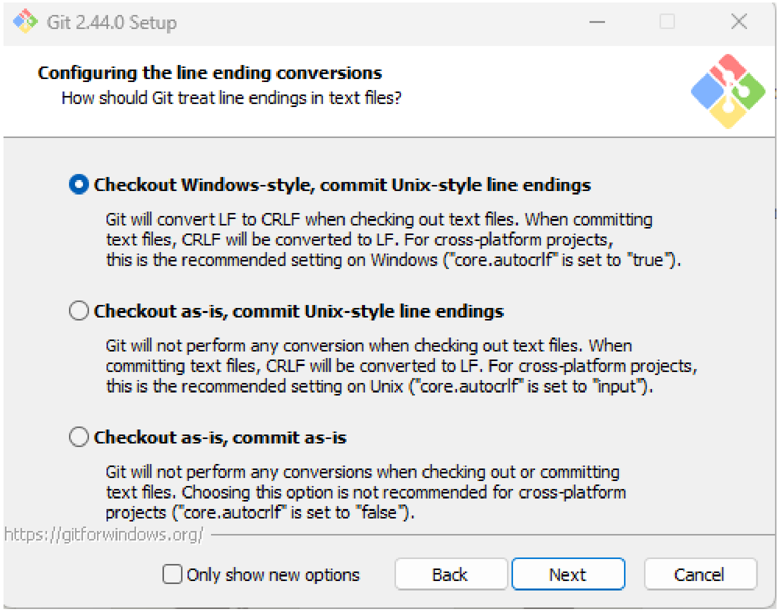
Server certificates will be validated using the ca-bundle.crt file.

○ **Use the native Windows Secure Channel library**

Server certificates will be validated using Windows Certificate Stores. This option also allows you to use your company's internal Root CA certificates distributed e.g. via Active Directory Domain Services.

https://gitforwindows.org/

8. Select Checkout Window-style, commit Unix-style line endings, then click next



Git 2.44.0 Setup

Configuring the line ending conversions

How should Git treat line endings in text files?

● **Checkout Windows-style, commit Unix-style line endings**

Git will convert LF to CRLF when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Windows ("core.autocrlf" is set to "true").

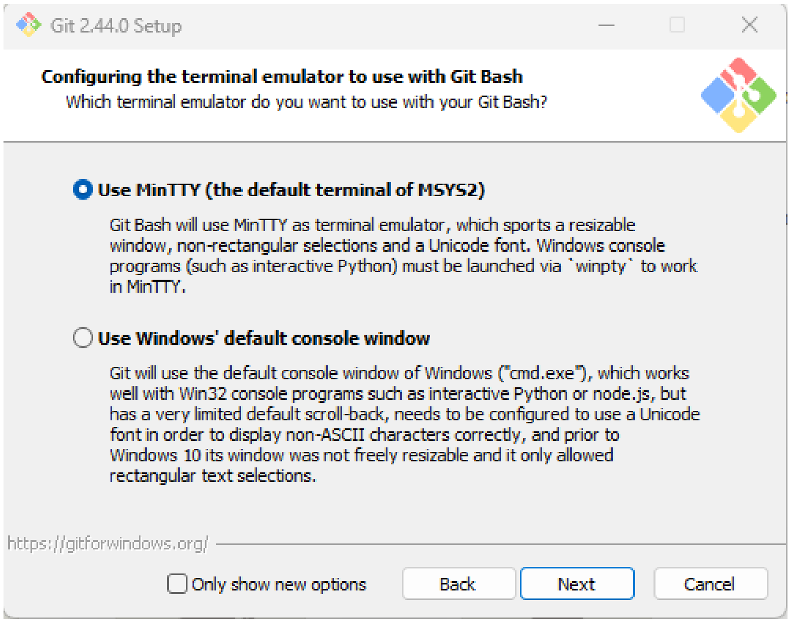
○ **Checkout as-is, commit Unix-style line endings**

Git will not perform any conversion when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Unix ("core.autocrlf" is set to "input").

○ **Checkout as-is, commit as-is**

Git will not perform any conversions when checking out or committing text files. Choosing this option is not recommended for cross-platform projects ("core.autocrlf" is set to "false").

9. Select Use MinTTY (the default terminal of MSYS2), then click next



Git 2.44.0 Setup

Configuring the terminal emulator to use with Git Bash

Which terminal emulator do you want to use with your Git Bash?

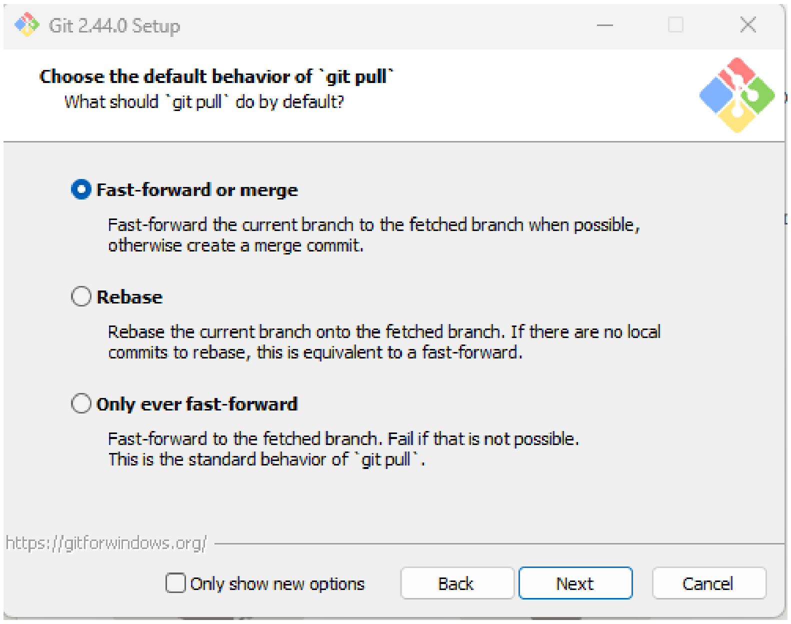
● **Use MinTTY (the default terminal of MSYS2)**

Git Bash will use MinTTY as terminal emulator, which sports a resizable window, non-rectangular selections and a Unicode font. Windows console programs (such as interactive Python) must be launched via `winpty` to work in MinTTY.

○ **Use Windows' default console window**

Git will use the default console window of Windows ("cmd.exe"), which works well with Win32 console programs such as interactive Python or node.js, but has a very limited default scroll-back, needs to be configured to use a Unicode font in order to display non-ASCII characters correctly, and prior to Windows 10 its window was not freely resizable and it only allowed rectangular text selections.

10. Select Fast-forward or merge, then click next



Choose the default behavior of `git pull`

What should `git pull` do by default?

● **Fast-forward or merge**

Fast-forward the current branch to the fetched branch when possible, otherwise create a merge commit.

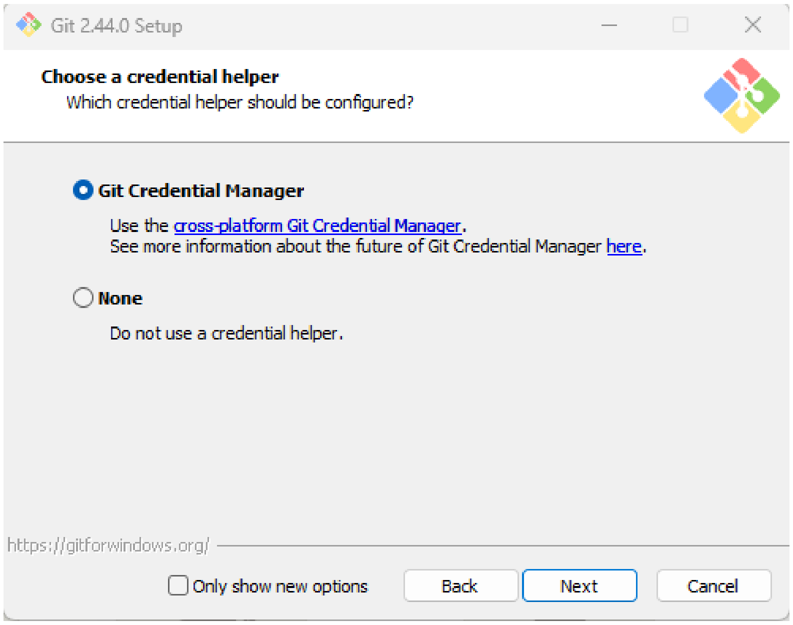
○ **Rebase**

Rebase the current branch onto the fetched branch. If there are no local commits to rebase, this is equivalent to a fast-forward.

○ **Only ever fast-forward**

Fast-forward to the fetched branch. Fail if that is not possible. This is the standard behavior of `git pull`.

11. Select Git Credential Manager, then click next



Choose a credential helper

Which credential helper should be configured?

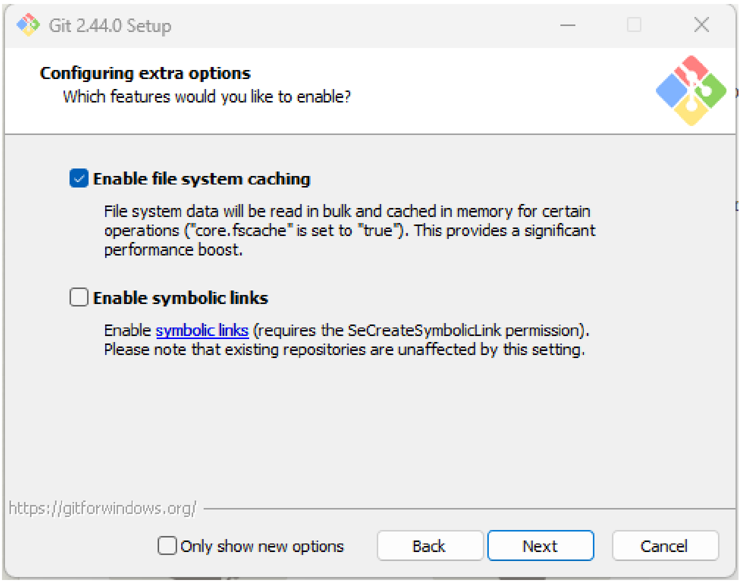
● **Git Credential Manager**

Use the cross-platform Git Credential Manager.
See more information about the future of Git Credential Manager here.

○ **None**

Do not use a credential helper.

12. Select Enable file system caching, then click next



Git 2.44.0 Setup

Configuring extra options

Which features would you like to enable?

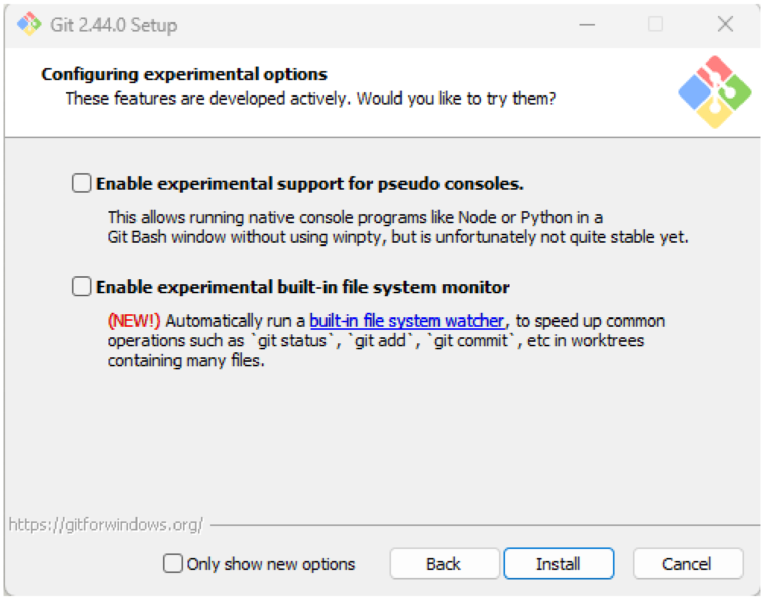
☑ **Enable file system caching**

File system data will be read in bulk and cached in memory for certain operations ("core.fscache" is set to "true"). This provides a significant performance boost.

☐ **Enable symbolic links**

Enable symbolic links (requires the SeCreateSymbolicLink permission). Please note that existing repositories are unaffected by this setting.

13. In Configuring experimental options window, no need to select any choice and click install



Git 2.44.0 Setup

Configuring experimental options

These features are developed actively. Would you like to try them?

☐ **Enable experimental support for pseudo consoles.**

This allows running native console programs like Node or Python in a Git Bash window without using winpty, but is unfortunately not quite stable yet.

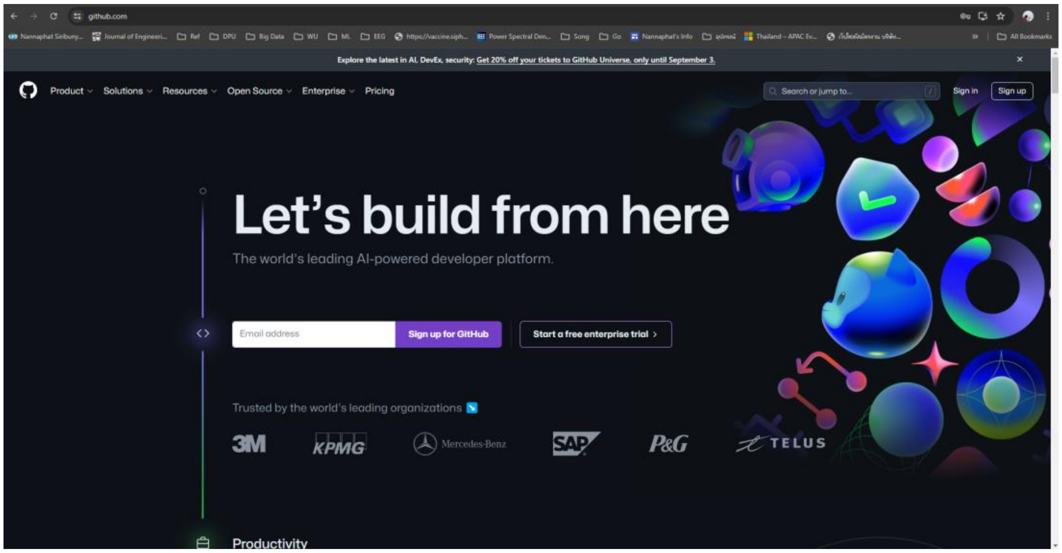
☐ **Enable experimental built-in file system monitor**

(NEW!) Automatically run a built-in file system watcher, to speed up common operations such as `git status`, `git add`, `git commit`, etc in worktrees containing many files.

14. Wait until install complete

Part 2: Create account Github

1. Go to website https://github.com/



Explore the latest in AI, DevEx, security: Get 20% off your tickets to GitHub Universe, only until September 3.

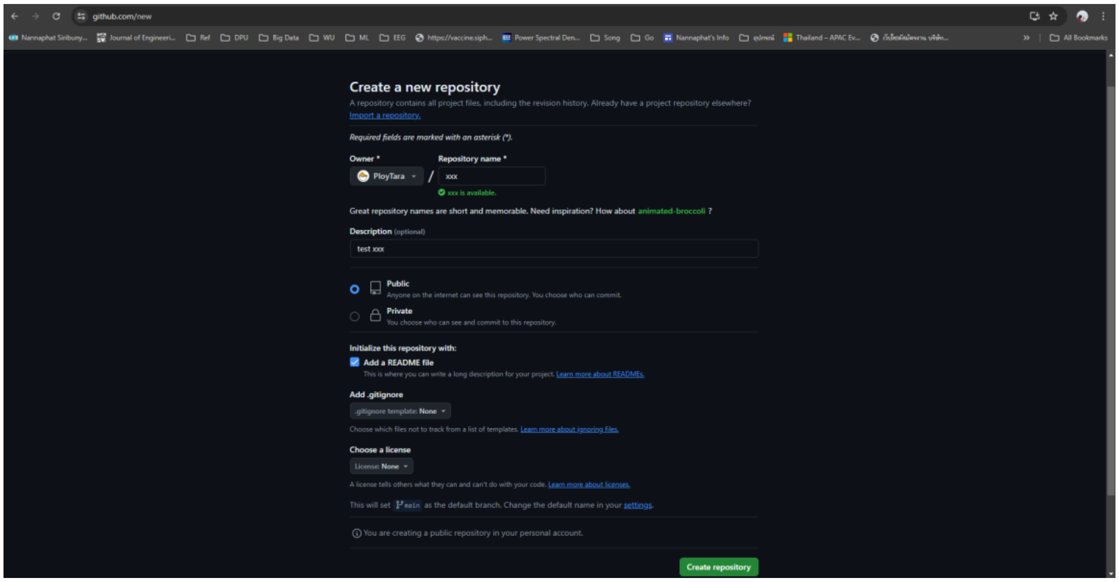
Let's build from here

The world's leading AI-powered developer platform.

Email address

Trusted by the world's leading organizations

2. Create your account then login to website
3. Click new button to create a new repository
4. In window New Repository, define your Repository name, add Description, select Public and Add a README file, the click Create repository



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Required fields are marked with an asterisk (*).

Owner * PloyTara / Repository name * xxx

xxx is available.

Great repository names are short and memorable. Need inspiration? How about animated-broccoli ?

Description (optional)

test xxx

● Public
Anyone on the internet can see this repository. You choose who can commit.

○ Private
You choose who can see and commit to this repository.

Initialize this repository with:

☑ Add a README file
This is where you can write a long description for your project. Learn more about READMEs.

Add .gitignore

.gitignore template: None

Choose which files not to track from a list of templates. Learn more about ignoring files.

Choose a license

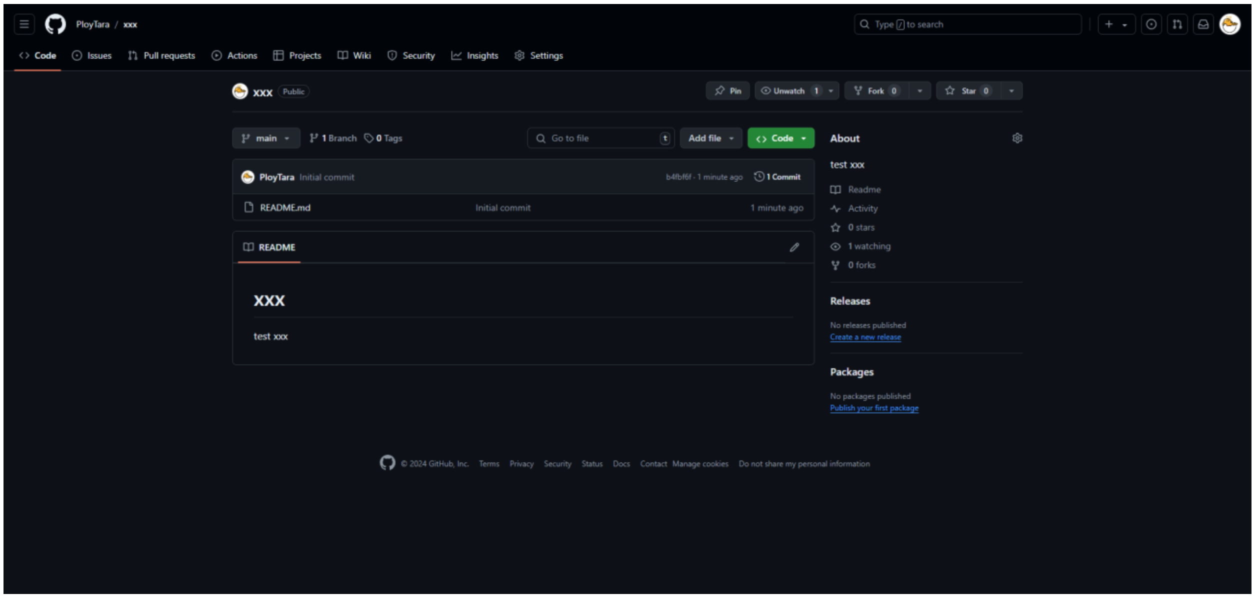
License: None

A license tells others what they can and can't do with your code. Learn more about licenses.

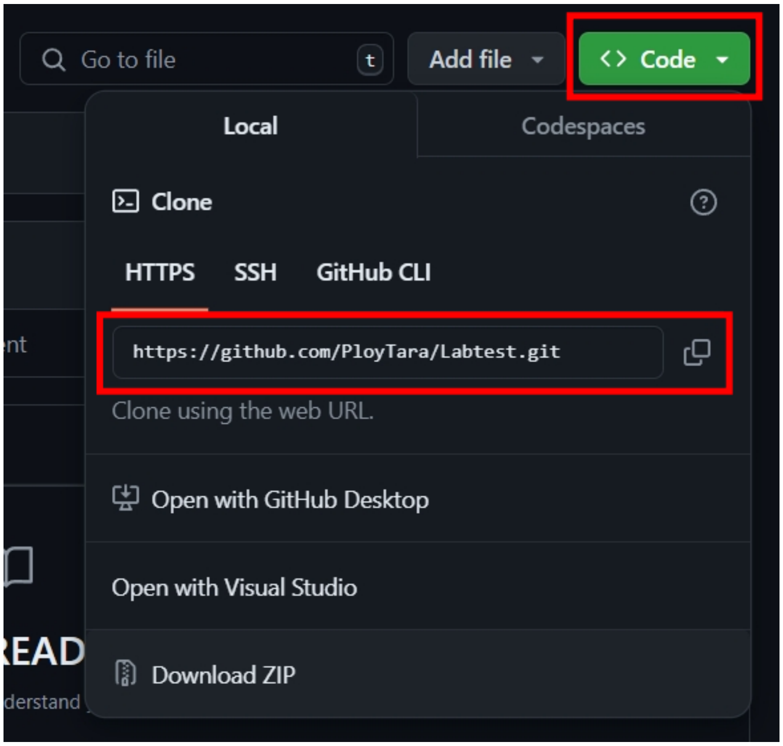
This will set `main` as the default branch. Change the default name in your settings.

You are creating a public repository in your personal account.

5. You will get the new repository in your account



6. You will get your Github URL by click at **<> Code** button



```
https://github.com/PloyTara/Labtest.git
```

Part 3: Create python file

1. Open Visual Studio Code, create your folder project and add file printName.py in this folder
2. Write code to print your student ID and your name and save

Your code in python file will be

Part 4: Bring python file to Github

1. Open terminal in Visual Studio Code
2. Use Git command in terminal
    a. Set up username and email
        i. git config –global user.name “username”
        ii. git config –global user.email “email”
    b. Initial Git
        i. cd your_workspace
        ii. git init
    c. Add file
        i. git add .

    d. Commit file

        i. git commit -m “your comment”

    e. Remote to Github

        i. git remote add <projectName_in_Github> <URL>

    f. Push new project to Github

        i. git push –set-upstream <projectName_in_Github> master

3. Check in your Github. What is happen in your Github?

----

4. Add comment in your file in VS code
5. Add your modified file to your Github by Git command in terminal

    a. Add modified file

        i. git add .

    b. Commit file

        i. git commit -m “your comment”

    c. Push updated project to your Github

        i. git push

6. Check in your Github. What is happen in your Github?

Part 5: Create new branch and push new branch to your Github

1. Create new branch
    a. git checkout -b “newBranchName”
2. Check at VS code, what is changing in your VS code?

3. Add new branch to your Github
    a. git push –set-upstream < projectName_in_Github > <newBranch>
4. Check in your Github. What is happen in your Github?

Part 6: Clone another file to own computer

1. Poke a friend and ask for their Github link.
2. Create your new directory and use this to pull your friend’s file
3. Open your VS code and close the previous project
4. Open your new directory by Open Folder…
5. Open terminal and clone your friend’s file
    a. git clone <your_friend_Github_URL>
6. Check at VS code, what is changing in your VS code?