



Programming Fundamentals

Programming Day - Week 01



Introduction

Welcome to your first programming day. In this lab manual, we shall work together to learn and implement new concepts including setting up a GitHub Account and using GitBash to remotely store your files.

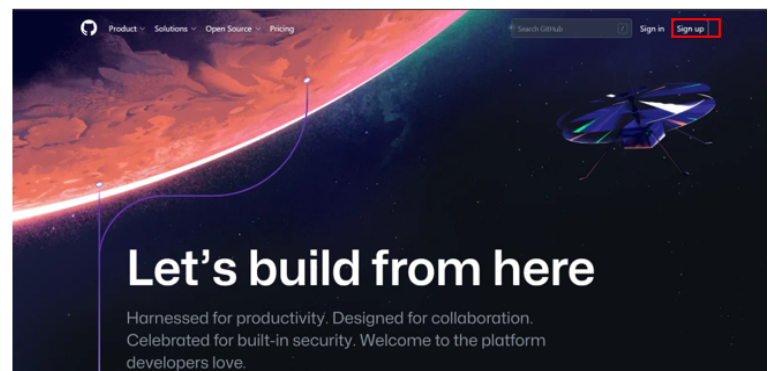
Skills to be learned:

- Creating a GitHub account and Installing GitBash to remotely store the files
- Adding, Committing, and Pushing files remotely to the GitHub repositories

Let's do some coding.

First Thing First, Let's create a Github Account.

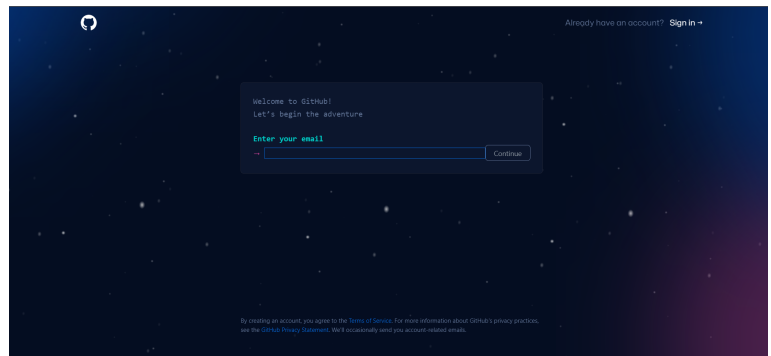
Go to <https://github.com/> and Click on **Sign Up** Button



You will need an email account for creating your GitHub account.

Enter the email address of your account.
Choose a password and **complete the profile**.

Note: In case you already have a GitHub Account, just sign in using that account.





Programming Fundamentals

Programming Day - Week 01



After that you will be asked to verify that you are a human

Would you like to receive product updates and announcements via email?

Type "y" for yes or "n" for no

✓ n

Verify your account

Protecting your account

Please solve this puzzle so we know you are a real person

Verify



After successful Verification, you will be given the launch code on your email that you chose to create a Github account with.

See your email and enter the launch code.

You're almost done!
We sent a launch code to maaida.mirza@gmail.com

→ Enter code*

Didn't get your email? [Resend the code](#) or [update your email address](#).


You will be asked to enter some information related to you

Question: How many team members are working with you?

Choose 2-5

Question: Are you a teacher or student?

Choose Student

 **Welcome to GitHub**
We are glad you're here.

This will help us guide you to the tools that are best suited for your projects.

How many team members will be working with you?

☐ Just me ☐ 2-5 ☐ 5-10

☐ 10-20 ☐ 20-50 ☐ 50+

Are you a student or teacher?

☐ N/A ☐ Student ☐ Teacher

[Continue](#)



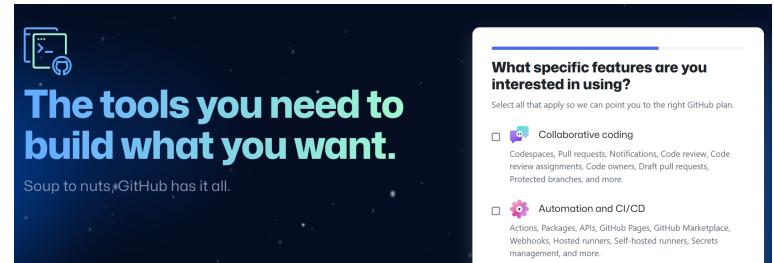
Programming Fundamentals

Programming Day - Week 01



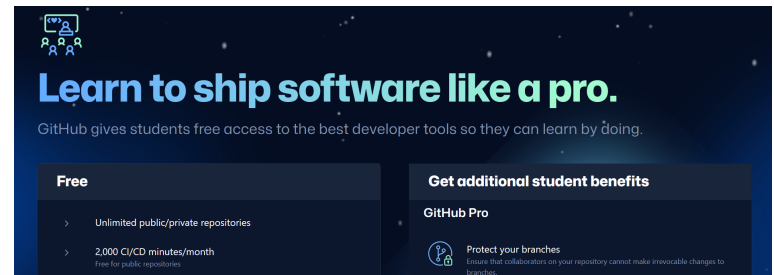
You will be asked to choose the features you are interested in.

Choose Collaborative Coding and and project management

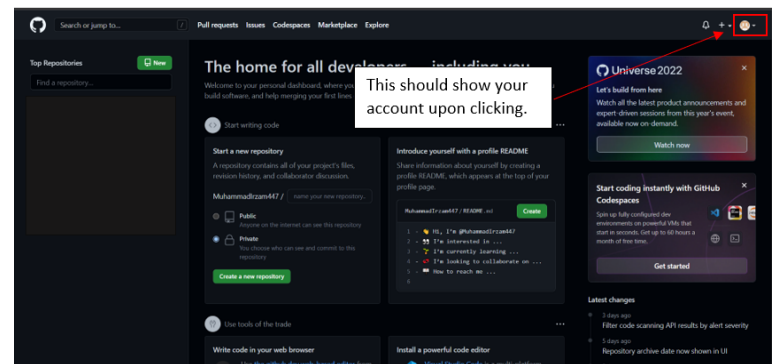


Then you will be asked to choose which version you want to use

Choose Continue for Free



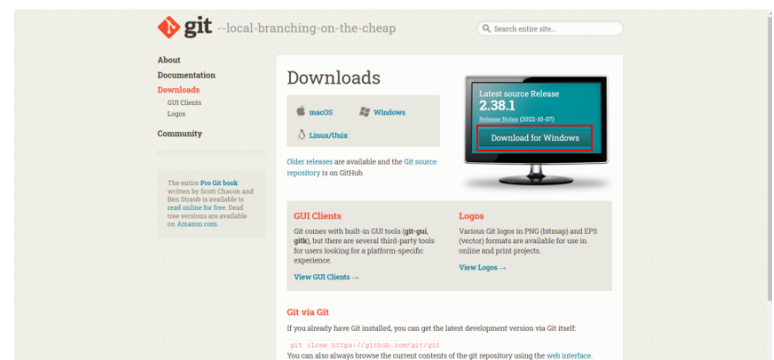
Welcome to your GitHub Account.



Now Let's Install GitBash.

Visit
<https://git-scm.com/download/>

To download GitBash which will be used for uploading files to your git account.



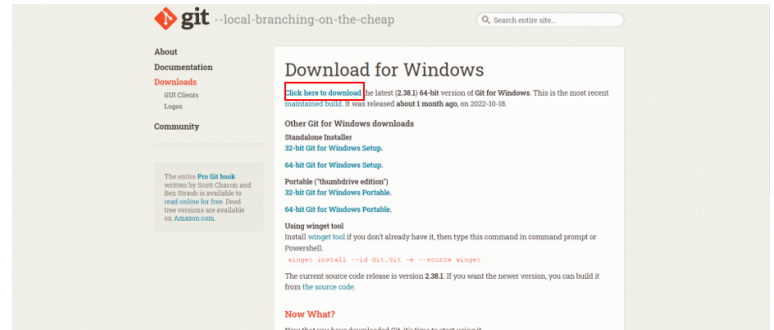


Programming Fundamentals

Programming Day - Week 01

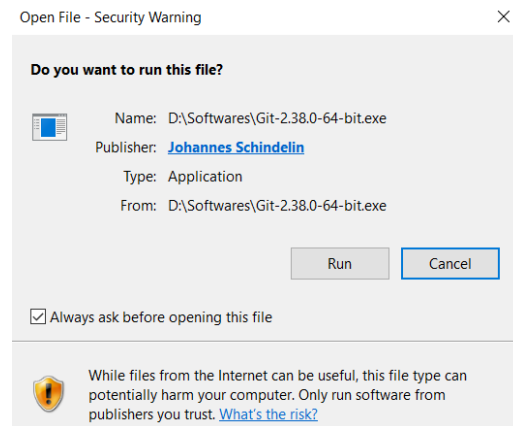


Install the Latest 64bit Setup for Windows.

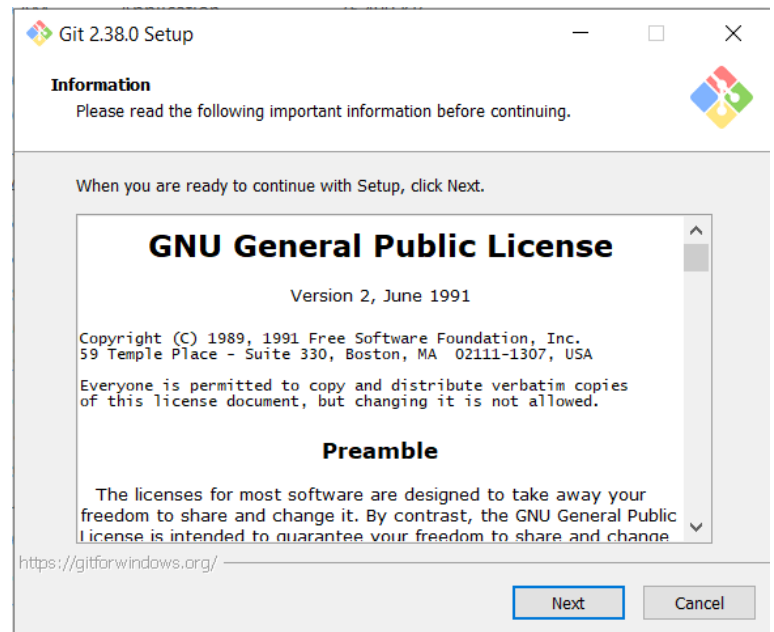


Sit tight, it may take a minute to complete the download.

Upon successful completion, Double Click on the downloaded file and choose **Run** from the menu.



Choose Next



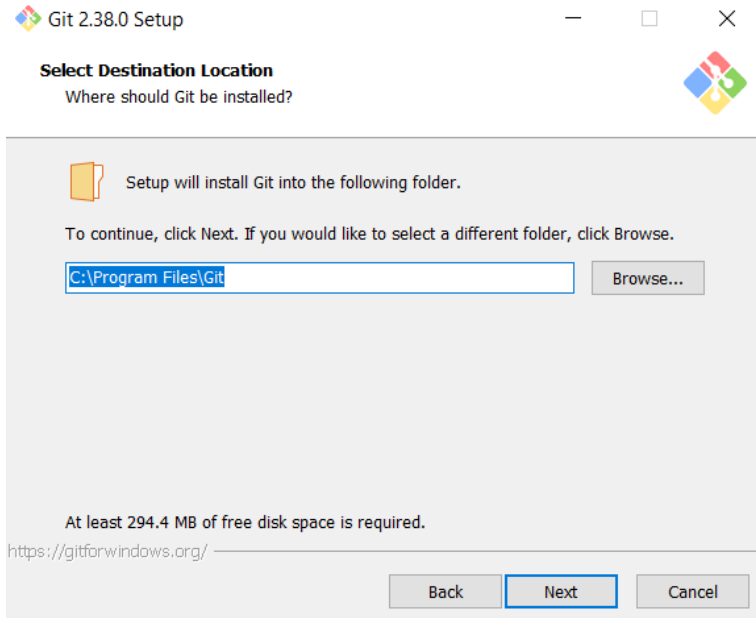


Programming Fundamentals

Programming Day - Week 01



Choose Next



Git 2.38.0 Setup

Select Destination Location
Where should Git be installed?

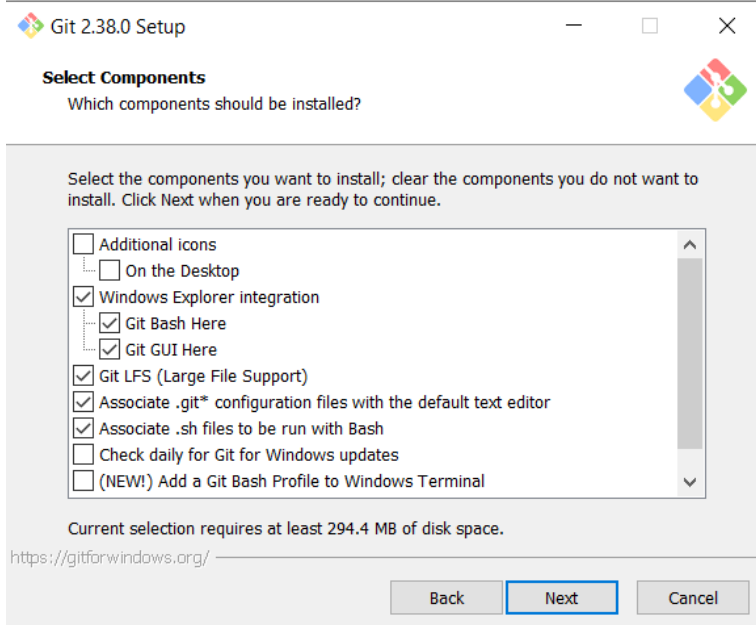
Setup will install Git into the following folder.

To continue, click Next. If you would like to select a different folder, click Browse.

At least 294.4 MB of free disk space is required.

<https://gitforwindows.org/>

Choose Next



Git 2.38.0 Setup

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
 - ☒ Git Bash Here
 - ☒ 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 294.4 MB of disk space.

<https://gitforwindows.org/>



Programming Fundamentals

Programming Day - Week 01



Choose **Next**

Git 2.38.0 Setup

Select Start Menu Folder
Where should Setup place the program's shortcuts?

Setup will create the program's shortcuts in the following Start Menu folder.

To continue, click Next. If you would like to select a different folder, click Browse.

☐ Don't create a Start Menu folder

<https://gitforwindows.org/>

Choose **Next**

Git 2.38.0 Setup

Choosing the default editor used by Git
Which editor would you like Git to use?

Use Vim (the ubiquitous text editor) as Git's default editor

The [Vim editor](#), while powerful, [can be hard to use](#). Its user interface is unintuitive and its key bindings are awkward.

Note: Vim is the default editor of Git for Windows only for historical reasons, and it is highly recommended to switch to a modern GUI editor instead.

Note: This will leave the 'core.editor' option unset, which will make Git fall back to the 'EDITOR' environment variable. The default editor is Vim - but you may set it to some other editor of your choice.

<https://gitforwindows.org/>

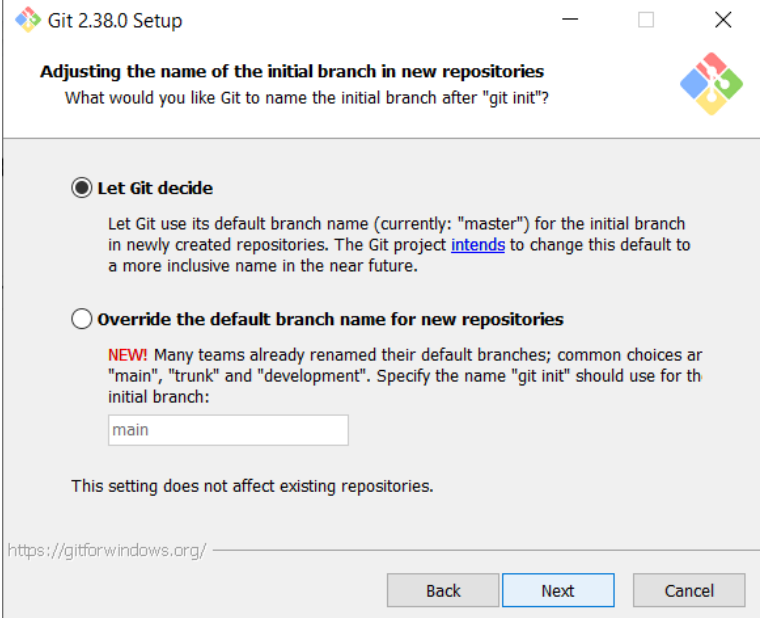


Programming Fundamentals

Programming Day - Week 01



Select **Let Git decide** and Choose **Next**



Git 2.38.0 Setup

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.

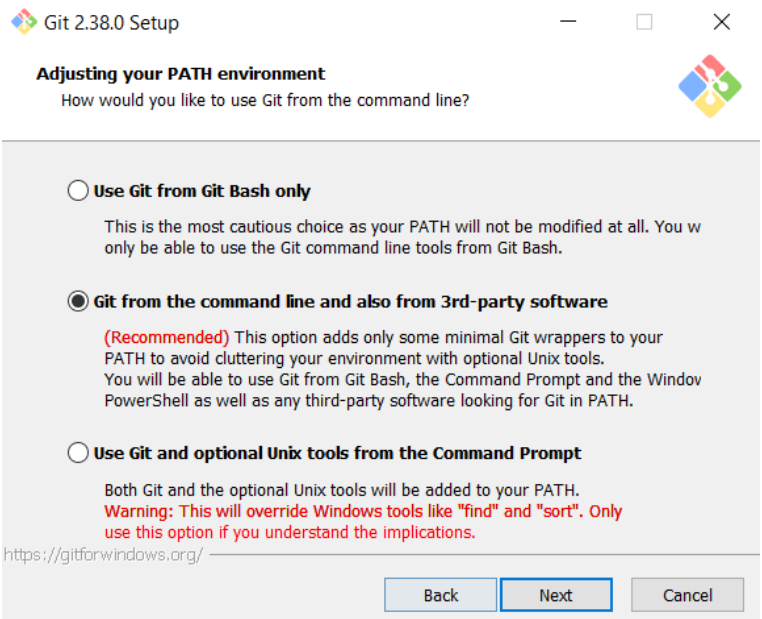
☐ **Override the default branch name for new repositories**
NEW! Many teams already renamed their default branches; common choices are "main", "trunk" and "development". Specify the name "git init" should use for the initial branch:

This setting does not affect existing repositories.

<https://gitforwindows.org/>

Back Next Cancel

Choose **Next**



Git 2.38.0 Setup

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/>

Back Next Cancel



Programming Fundamentals

Programming Day - Week 01



Choose **Next**

Git 2.38.0 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/>

Back Next Cancel

Choose **Next**

Git 2.38.0 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/>

Back Next Cancel



Programming Fundamentals

Programming Day - Week 01



Choose Next

Git 2.38.0 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").

<https://gitforwindows.org/>

Back Next Cancel

Choose Next

Git 2.38.0 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 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.

<https://gitforwindows.org/>

Back Next Cancel



Programming Fundamentals

Programming Day - Week 01



Choose Next

Git 2.38.0 Setup

Choose the default behavior of `git pull`

What should `git pull` do by default?

☒ **Default (fast-forward or merge)**

This is the standard behavior of `git pull`: 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.

<https://gitforwindows.org/>

Back

Next

Cancel

Choose Next

Git 2.38.0 Setup

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.

<https://gitforwindows.org/>

Back

Next

Cancel



Programming Fundamentals

Programming Day - Week 01



Choose Next

Git 2.38.0 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.

<https://gitforwindows.org/>

Back Next Cancel

Choose Next

Git 2.38.0 Setup

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

☐ **Enable experimental support for pseudo consoles.**
(NEW!) This allows running native console programs like Node or Python in a Git Bash window without using winpty, but it still has known bugs.

☐ **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.

<https://gitforwindows.org/>

Back Install Cancel

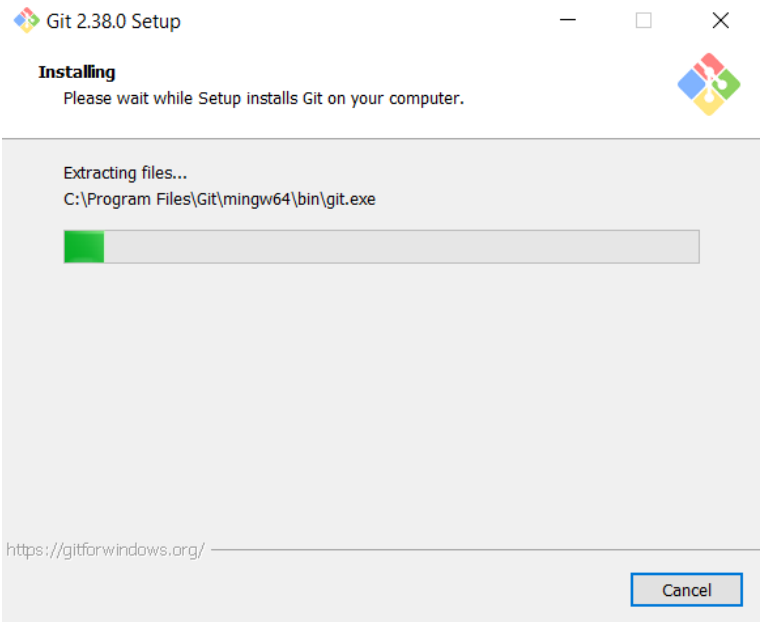


Programming Fundamentals

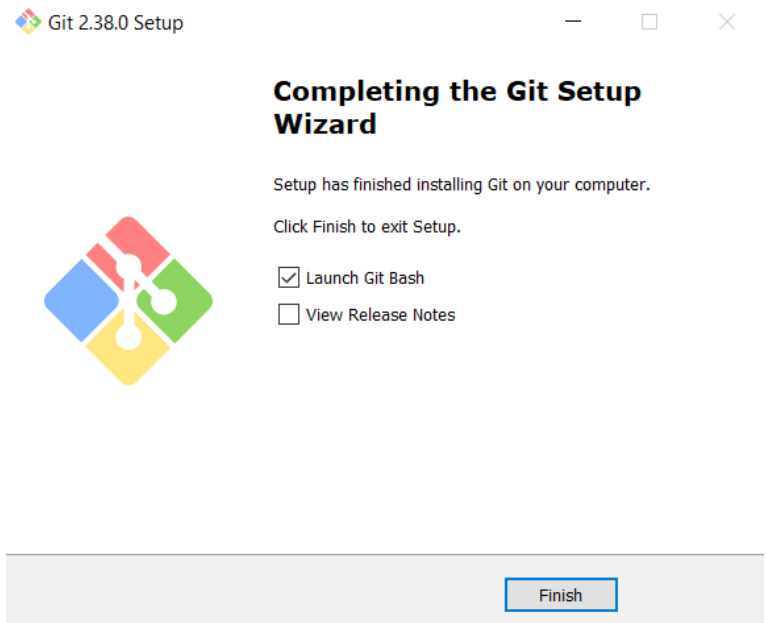
Programming Day - Week 01



Please Wait for the setup to install.



Choose **Finish**



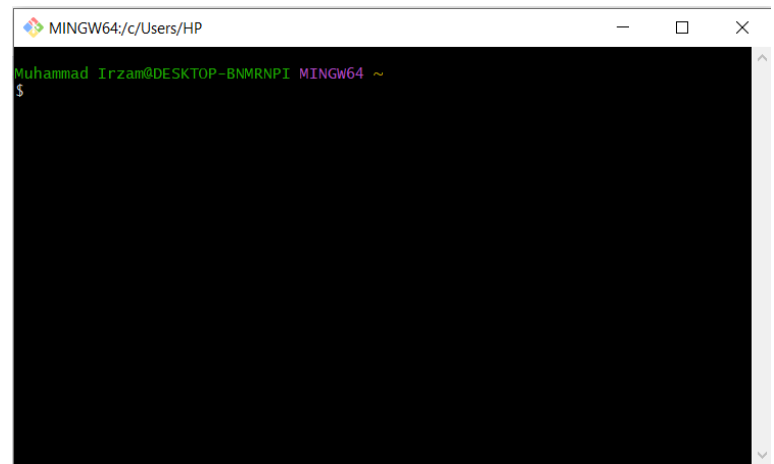


Programming Fundamentals

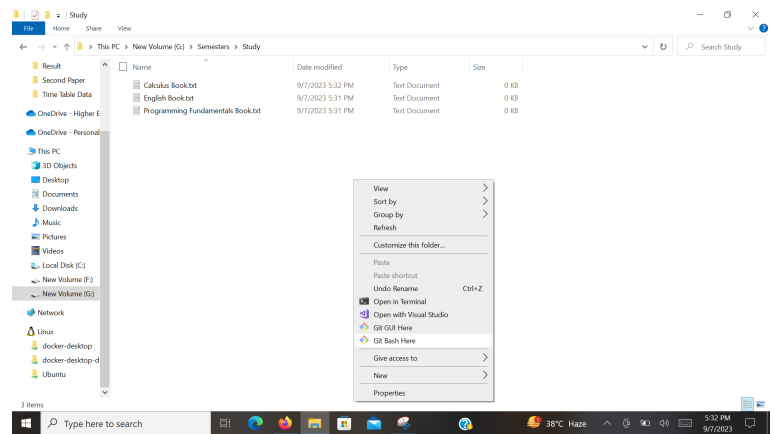
Programming Day - Week 01



GitBash cmd should appear on your screen upon successful completion.



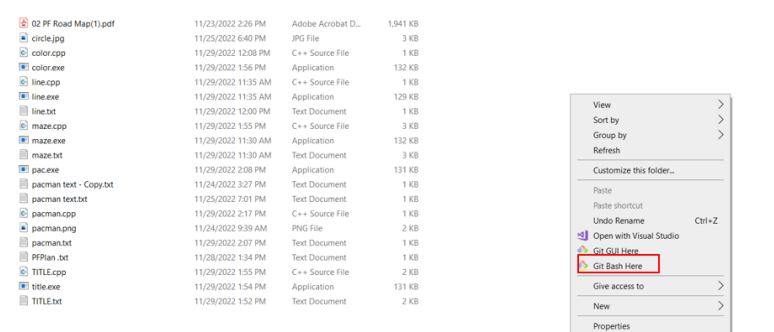
In order to open gitBash in a specific directory, right click on any empty/white space and choose **Git Bash Here**



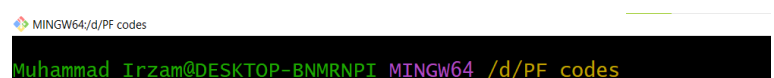
Great work. You have successfully installed GitBash.

Now, Let's get back and Configure the GitBash so it will be connected to your GitHub Account.

Navigate to the folder where you have stored your codes and Right Click in the window and choose **GitBash here**.



It will open the GitBash cmd in that directory





Programming Fundamentals

Programming Day - Week 01



| | |
|--|--|
| <p>Write the following command in the cmd.</p> <pre>git config --global user.name "yourGithubUserName"</pre> <p>Replace the username with your own GitHub username. Here is a working example:</p> | <pre>Hp@Laptop-285-180 MINGW64 /g/Semesters/Study \$ git config --global user.name "MaidaShahid1122"</pre> |
| <p>Write the following command in the cmd.</p> <pre>git config --global user.email "yourEmailAccount"</pre> <p>Replace the email account with your own email account that you are using for GitHub. Here is a working example:</p> | <pre>Hp@Laptop-285-180 MINGW64 /g/Semesters/Study \$ git config --global user.email "maaida.mirza@gmail.com"</pre> |
| <p>Note: These commands are a one-time task only and all the remaining steps are repeated for each new directory(folder) that you want to upload to your GitHub account.</p> | |
| <p>git init</p> <p>Execute this command, to start your session.</p> <p>Note: The good thing is you need to execute this command only once to start your session in a specific directory.</p> | <pre>Muhammad Irzam@DESKTOP-BNMRNPI MINGW64 /d/PF codes \$ git init Initialized empty Git repository in D:/PF codes/.git/</pre> |
| <p>git add .</p> <p>This command is used to add all the files that are in your current working repository.</p> | <pre>Muhammad Irzam@DESKTOP-BNMRNPI MINGW64 /d/PF codes (master) \$ git add .</pre> |
| <p>git commit -m "message"</p> <p>This command is used to make a commit (time stamp record of your file and changes in that) in your Git repository.</p> <p>NOTE: Now, do not close this window and go to the web browser.</p> | <pre>Muhammad Irzam@DESKTOP-BNMRNPI MINGW64 /d/PF codes (master) \$ git commit -m "message" [master (root-commit) 4e95968] message 3 files changed, 3 insertions(+) create mode 100644 file1.txt create mode 100644 file2.txt create mode 100644 file3.txt</pre> |



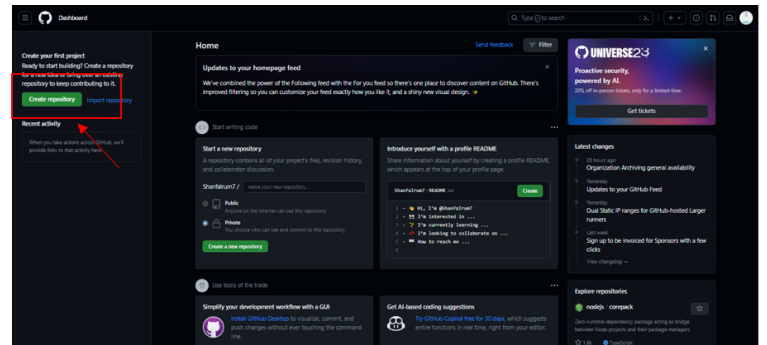
Programming Fundamentals

Programming Day - Week 01



Now go to your GitHub Account(www.github.com/) and create a Repository.

Look at the attached screen to create a repo.

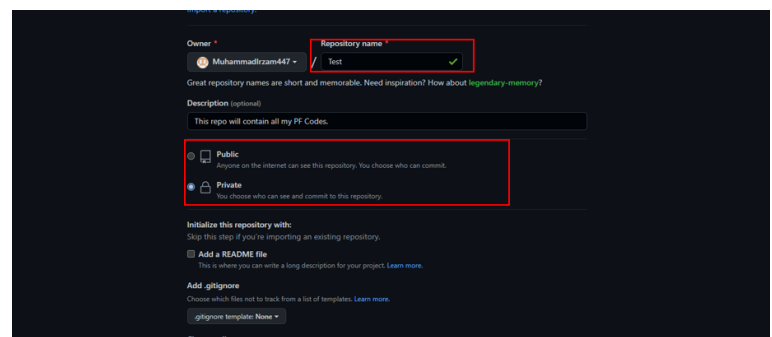


Give it a Name and Provide a description.

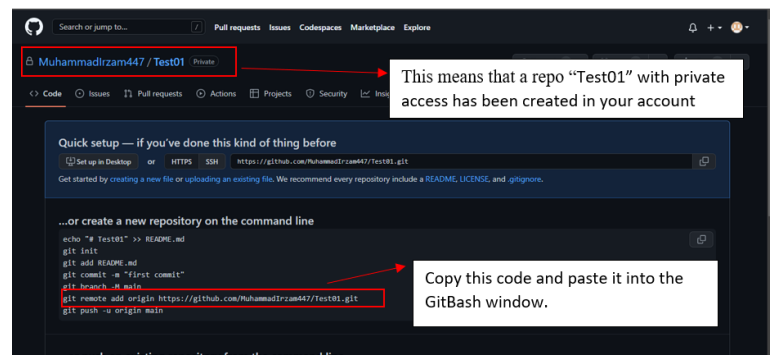
A **Public Repository** is accessed by anyone on the internet.

A **Private Repository** can only be accessed by you.

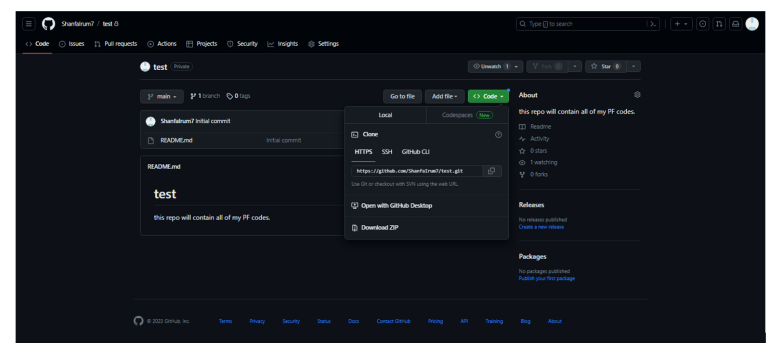
For now, choose **Private** and click on **Create Repository**.



On Successful creation, the attached screen should appear.



Maybe some get the following screen.



Now, **switch back to GitBash** and right click in the GitBash Window and Click on the **Paste** option and Hit





Programming Fundamentals

Programming Day - Week 01



Enter.

The command should look like this:

git remote add origin https://

Now, upload the files to the remote repo using the following command.

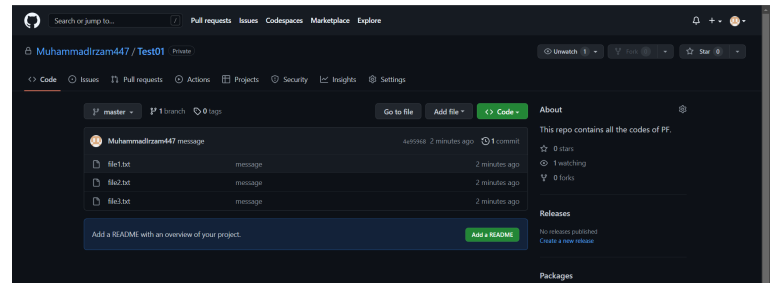
git push -u origin master

This command is used to upload all the committed files to your remote repository.

```
Muhammad Irzam@DESKTOP-BNMRNPI MINGW64 /d/PF codes (master)
$ git push -u origin master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (5/5), 315 bytes | 157.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/MuhammadIrzam447/Test01.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
```

Congratulations, you have created your GitHub Account and used GitBash to Upload the files to the remote repository.

You can check that by visiting your GitHub account and choosing your created repo.



Congratulations, you have successfully created your GitHub account and uploaded all the files in the PF Codes folder. Great Work Guys! You have just added another skill to your skill set.

Conclusion

Lets explain why uploading files to a GitHub account is beneficial:

Backup and Security:

Uploading your work to GitHub serves as a backup. Computers can crash, and files can get lost. By keeping your code on GitHub, it's safe and accessible from anywhere with an internet connection. Your work is secure even if something goes wrong with your local machine.

Showcasing Your Work:

GitHub is not only a tool for development but also a platform to showcase your skills to your TAs as well as to your juniors in the future. By sharing your projects on GitHub, you create a portfolio of your work. This can be a significant asset when applying for internships or jobs in the tech industry. Industry Employers often look at GitHub profiles to assess a candidate's coding abilities and commitment to the field.

Version Control and Collaboration:



Programming Fundamentals

Programming Day - Week 01



GitHub is a platform that helps you manage and keep track of changes in your code and files. Imagine you're working on a coding project, and you make a change that causes an issue. With GitHub, you can easily roll back to a previous, working version of your code. It acts like a time machine for your work, helping you avoid costly mistakes.

Moreover, if you're working on a team project, GitHub allows multiple people to collaborate seamlessly. It tracks who made which changes and when, making teamwork much smoother. This is crucial in the software development industry, where collaboration and version control are essential.

Documentation and Project Management:

GitHub provides tools for project management, issue tracking, and documentation. This is handy for keeping your projects organized and well-documented, which is essential for large or complex projects.

Summary of the commands used are given below:

| Command | Description |
|--|--|
| git config --global user.email "email" | It's a one time use only command, that is used to connect the GitBash with your GitHub Account |
| git config --global user.name "name" | This is a one time use only command that is used to connect the user remotely with their GitHub Account. |
| git init | This is used to initiate the session. It is used every time you want to create a new repo on the GitHub account. |
| git add . | This command is used to add all the files in the current working directory to the list of files that are to be uploaded on the GitHub account. There are various other versions on this command that you can find out on your own. |
| git commit -m "m" | This command makes the commit along with the message that you want to associate with that commit. |
| git remote add origin https://... | This command associates all the committed files to the defined origin repo . |
| git push -u origin master | This command is used to upload all the committed files to the defined origin repo. |



Programming Fundamentals

Programming Day - Week 01



| | |
|------------------------------|--|
| Ctrl + L | This is used to clear the screen in GitBash |
| git ls-files | This lists all the files in the current local directory. |
| git clone https://... | This command is used to download the remote repo files to your personal computer. |

Task 01(OP): Upload all your NotePad files to a public repository on your GitHub account.

Good Luck and Best Wishes !!

Happy Coding ahead :)