Exp .No : 1. Basic Installation of GIT and GITHUB.

Aim : To install the GIT and GITHUB software.

What is Git?

''Git'' is a go-to-version control tool that allows developers to access all of the code and efficiently manage their source code and track file changes be it small, medium, or massive application development. It remains the most widely used open-source distributed version control system (DVCS) till date and has been in use for over a decade after its initial release.

Unlike other version control systems that store a project's full version history in one place, Git gives each developer their repository locally containing the entire history of changes and the entire application.

Git is used to tracking changes in the source code hence tracks history

Git is a distributed version control tool for source code management. It is free and open-source.

Git creates backup automatically, as the developer has a version of the code on the local repository.

Git allows multiple developers to work together; hence is scalable and supports collaboration.

Git supports non-linear development because of its thousands of parallel branches.

This Git cheat sheet can serve as a ready reckoner to essential git commands that you may need to use in your coding career.

Installing Git

Before you start using Git, you have to make it available on your computer. Even Though if it is installed, it is probably a good idea to update to the latest version. You can either install it as a package or via another installer or download the source code and compile it yourself.

Installing on Linux

If you want to install the essential Git tools on Linux via a binary installer, you can generally do so through the package management tool that comes with your distribution. If you are on operating systems such as Fedora, you can use dnf.

Git Cheat Sheet: Quick Guide for Reference

Installing on Linux	If you want to install the basic Git tools on Linux via a binary installer, you can generally do so through the package management tool that comes with your distribution. If you are on operating systems such as Fedora you can use dnf.	$ sudodnf install git-all
	If you are on a Debian-based distribution, such as Ubuntu use the	$ sudo apt install git-all

	following command	
Installing on macOS	There are several ways to install Git on a Mac. The easiest way is to install it from the Github website	https://mac.github.com
Installing on Windows	There are also a few ways to install Git on Windows. The most official build is available for download on the Git website.	https://git-scm.com/download/win

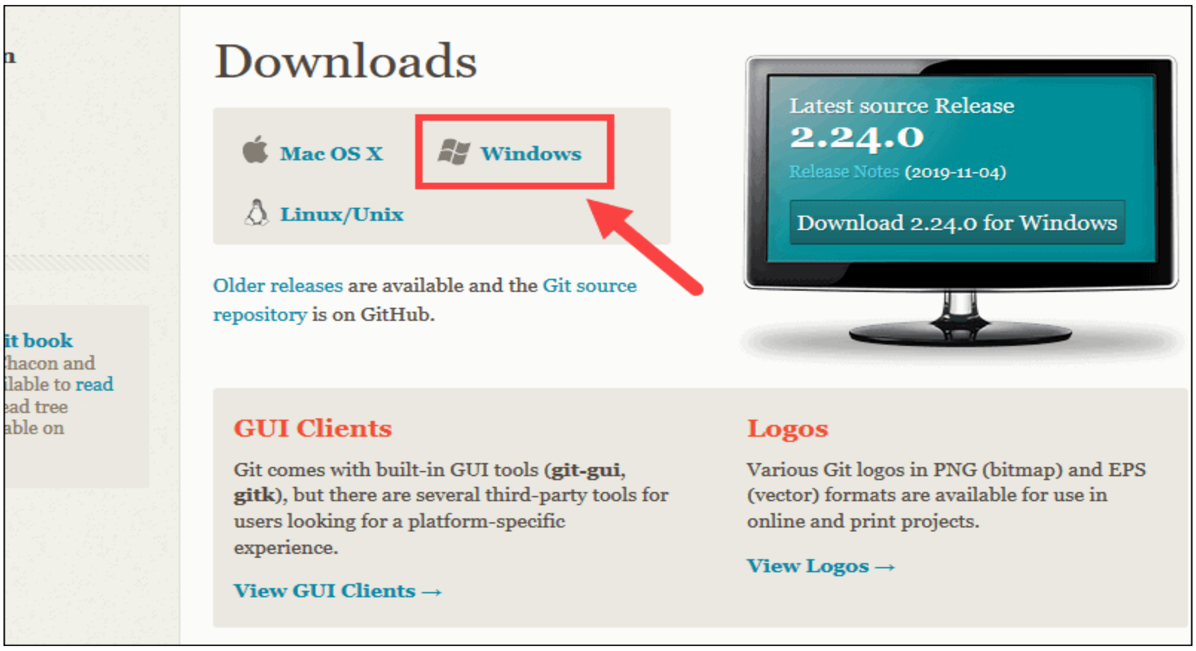
Steps For Installing Git for Windows

Installing Git prompts you to select a text editor. If you don't have one, we strongly advise you to install prior to installing Git. Our roundup of the best text editors for coding may help you decide.

Download Git for Windows

Browse to the official Git website: https://git-scm.com/downloads

Click the download link for Windows and allow the download to complete.



Older releases are available and the Git source repository is on GitHub.

GUI Clients

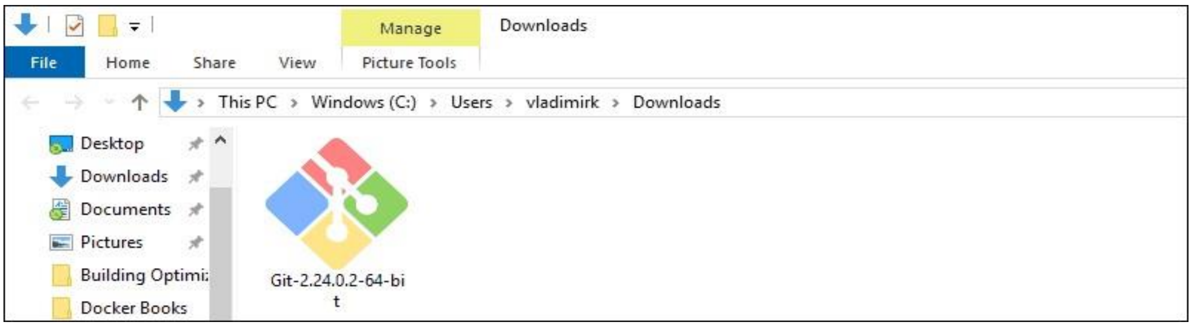
Git comes with built-in GUI tools (**git-gui**, **gitk**), but there are several third-party tools for users looking for a platform-specific experience.

Logos

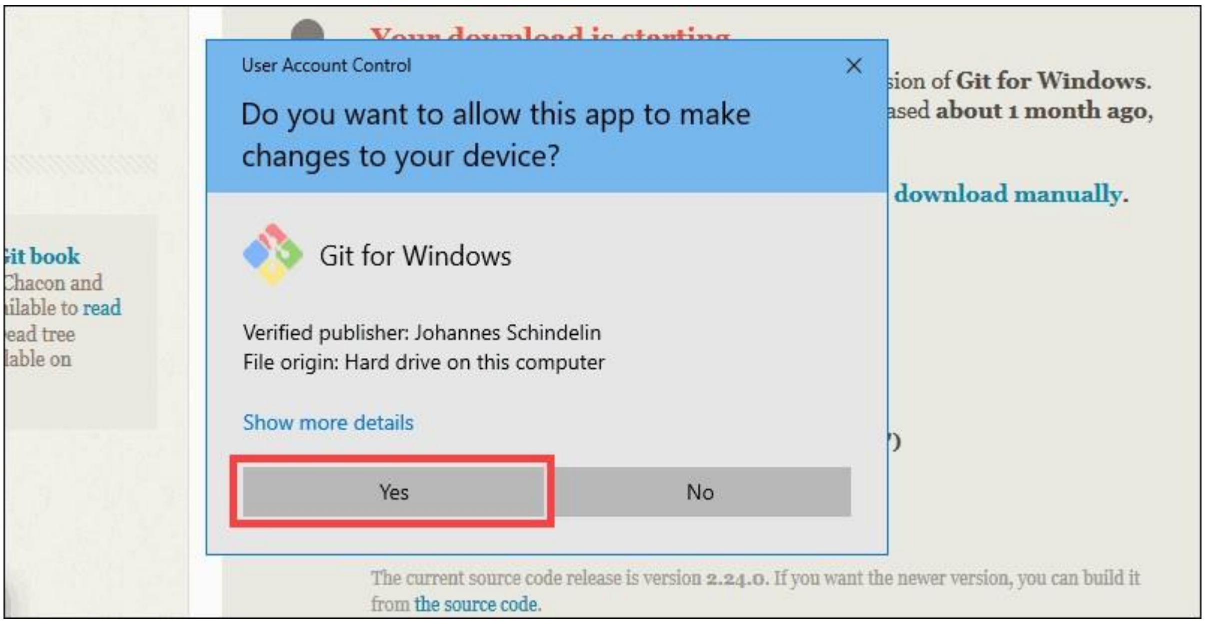
Various Git logos in PNG (bitmap) and EPS (vector) formats are available for use in online and print projects.

Extract and Launch Git Installer

Browse to the download location (or use the download shortcut in your browser). Double-click the file to extract and launch the installer.



Allow the app to make changes to your device by clicking Yes on the User Account Control dialog that opens.

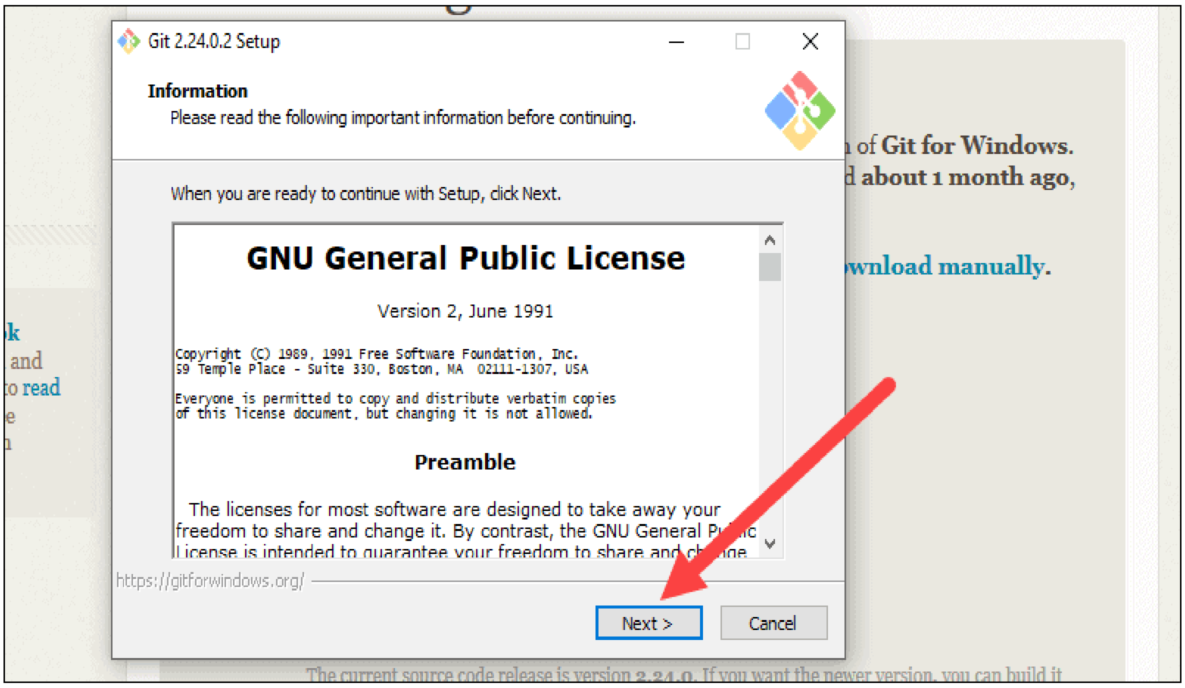

User Account Control

Do you want to allow this app to make changes to your device?

Git for Windows

Verified publisher: Johannes Schindelin
File origin: Hard drive on this computer

Review the GNU General Public License, and when you're ready to install, click Next.


Git 2.24.0.2 Setup

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

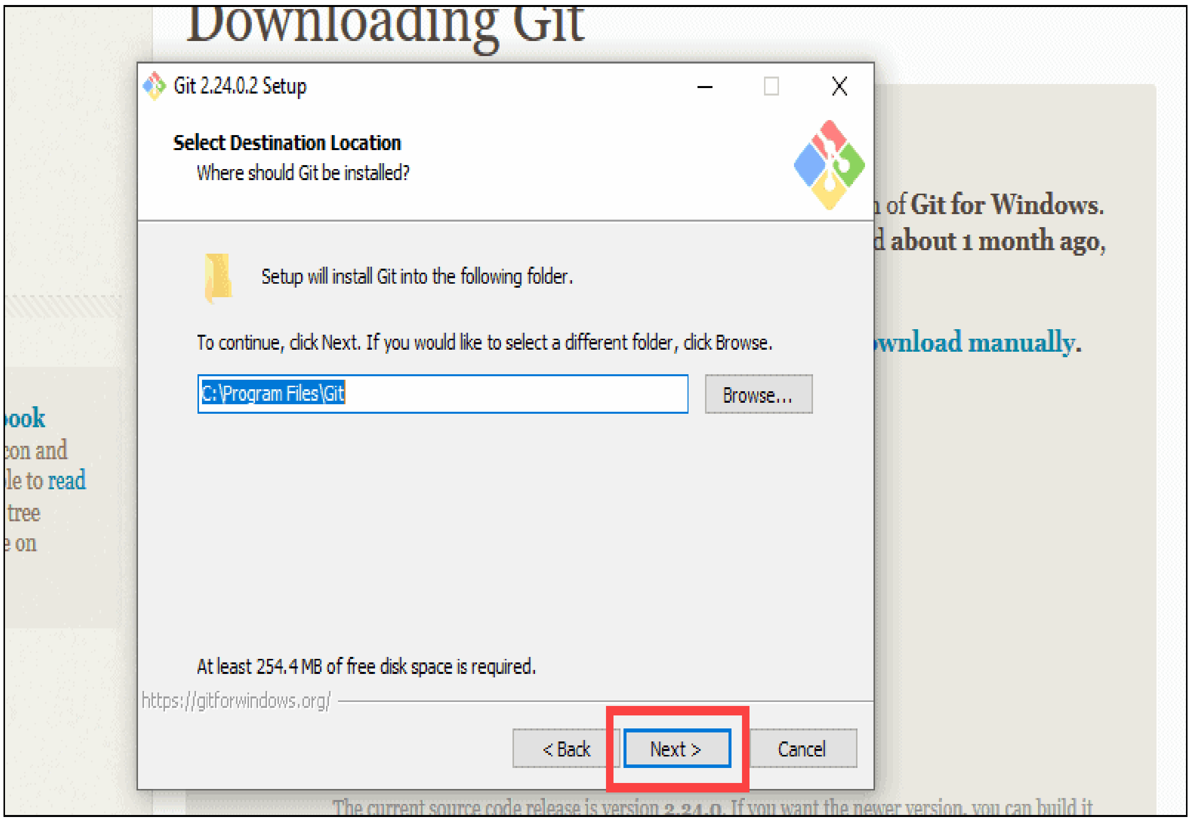
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

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/

The installer will ask you for an installation location. Leave the default, unless you have reason to change it, and click Next.


Git 2.24.0.2 Setup

Select Destination Location

Where should Git be installed?

Setup will install Git into the following folder.

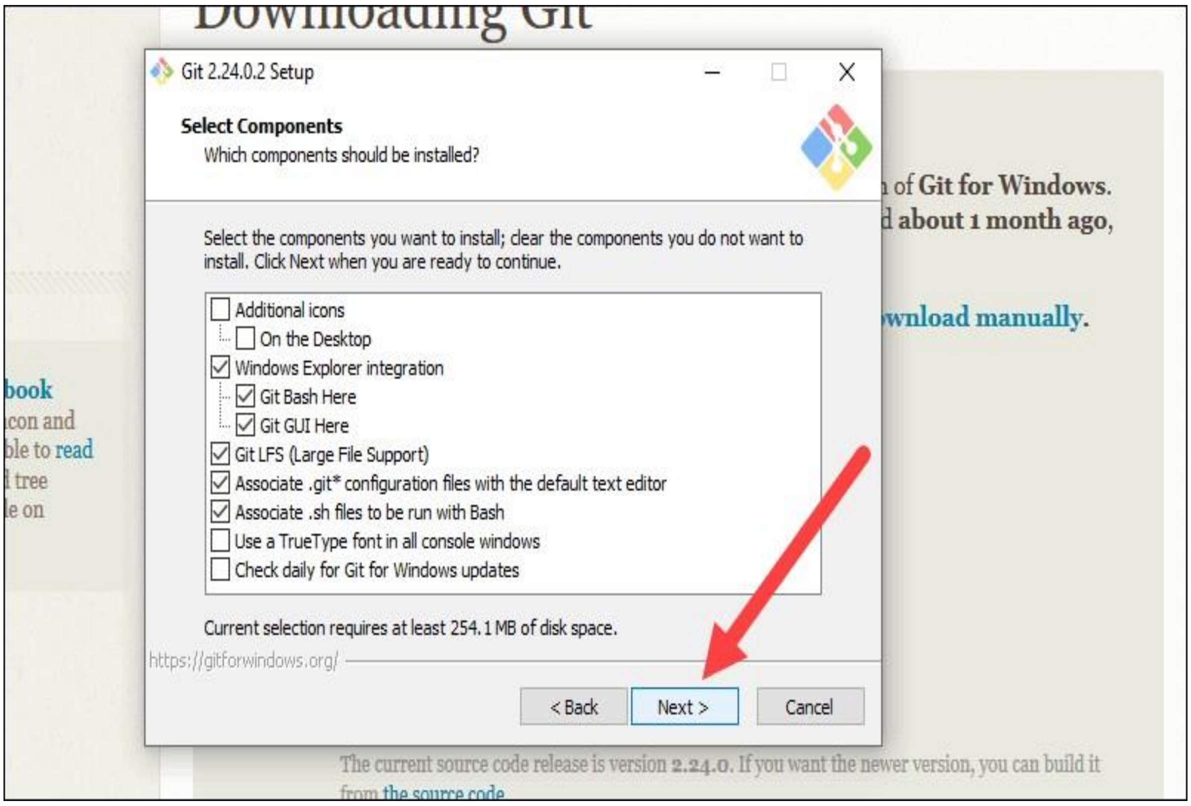
To continue, click Next. If you would like to select a different folder, click Browse.

C:\Program Files\Git

At least 254.4 MB of free disk space is required.

https://gitforwindows.org/

A component selection screen will appear. Leave the defaults unless you have a specific need to change them and click Next.


Git 2.24.0.2 Setup

Select Components

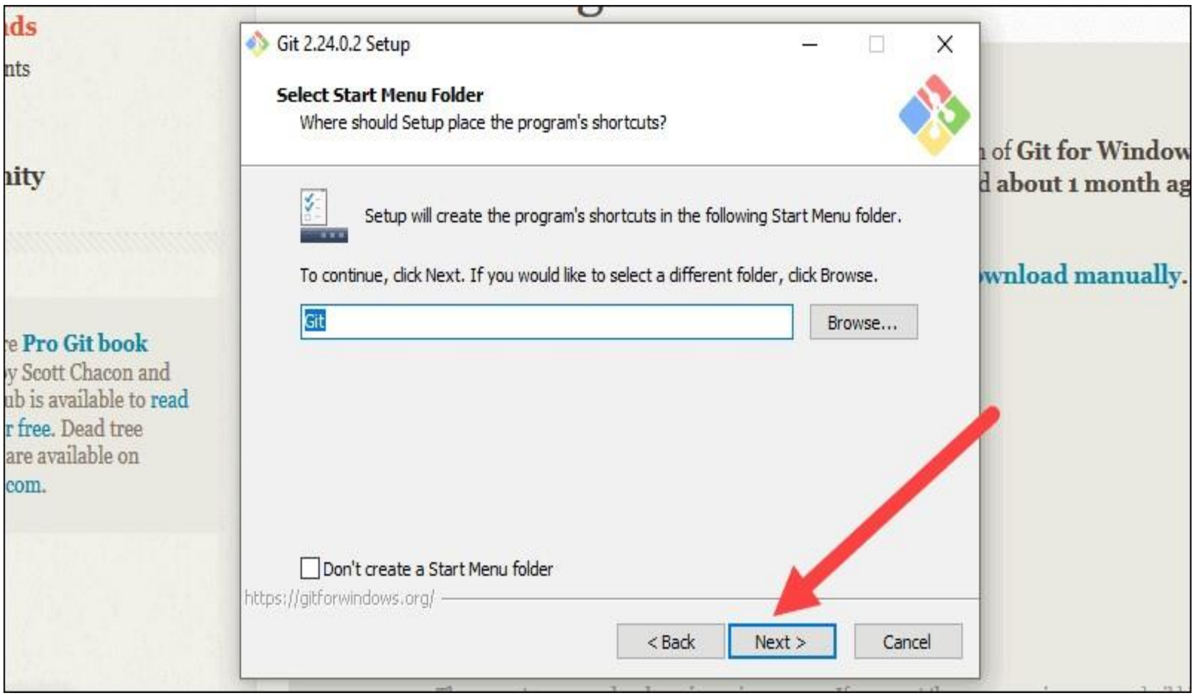
Which components should be installed?

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
- [x] Windows Explorer integration
  - [x] Git Bash Here
  - [x] Git GUI Here
- [x] Git LFS (Large File Support)
- [x] Associate .git* configuration files with the default text editor
- [x] Associate .sh files to be run with Bash
- [ ] Use a TrueType font in all console windows
- [ ] Check daily for Git for Windows updates

Current selection requires at least 254.1 MB of disk space.

The installer will offer to create a start menu folder. Simply click Next.


Git 2.24.0.2 Setup

Select Start Menu Folder

Where should Setup place the program's shortcuts?

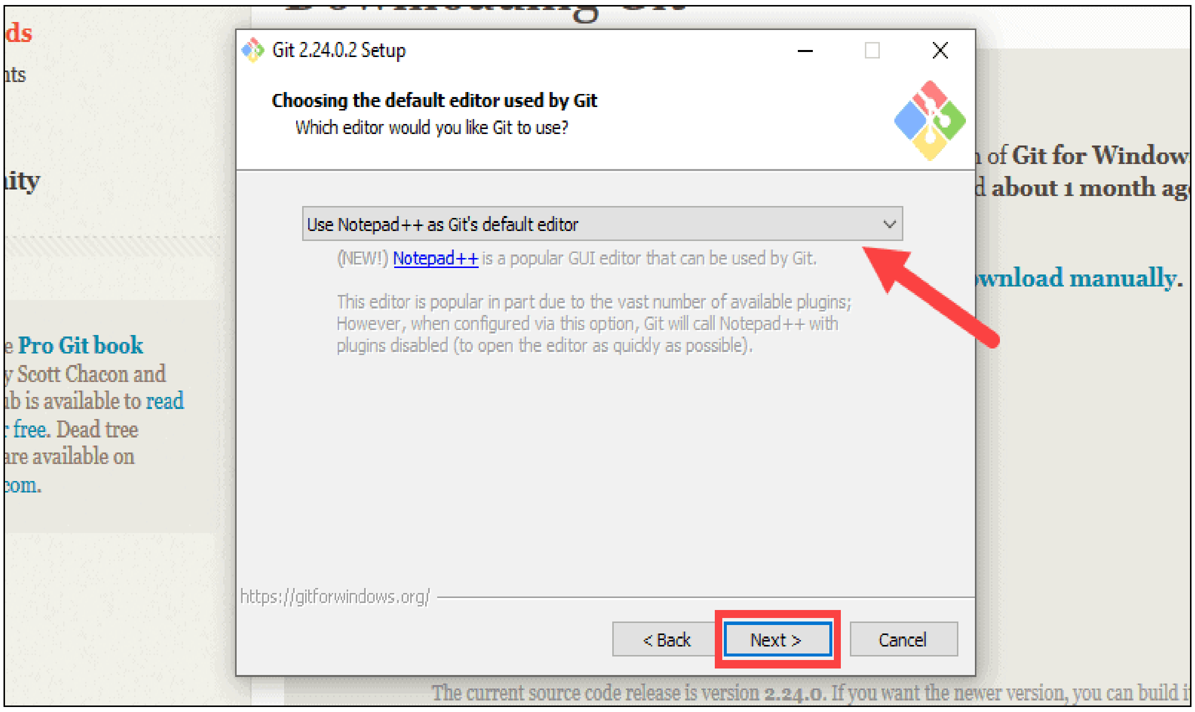
Setup will create the program's shortcuts in the following Start Menu folder.

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

Git

- [ ] Don't create a Start Menu folder

Select a text editor you'd like to use with Git. Use the drop-down menu to select Notepad++ (or whichever text editor you prefer) and click Next.



Choosing the default editor used by Git

Which editor would you like Git to use?

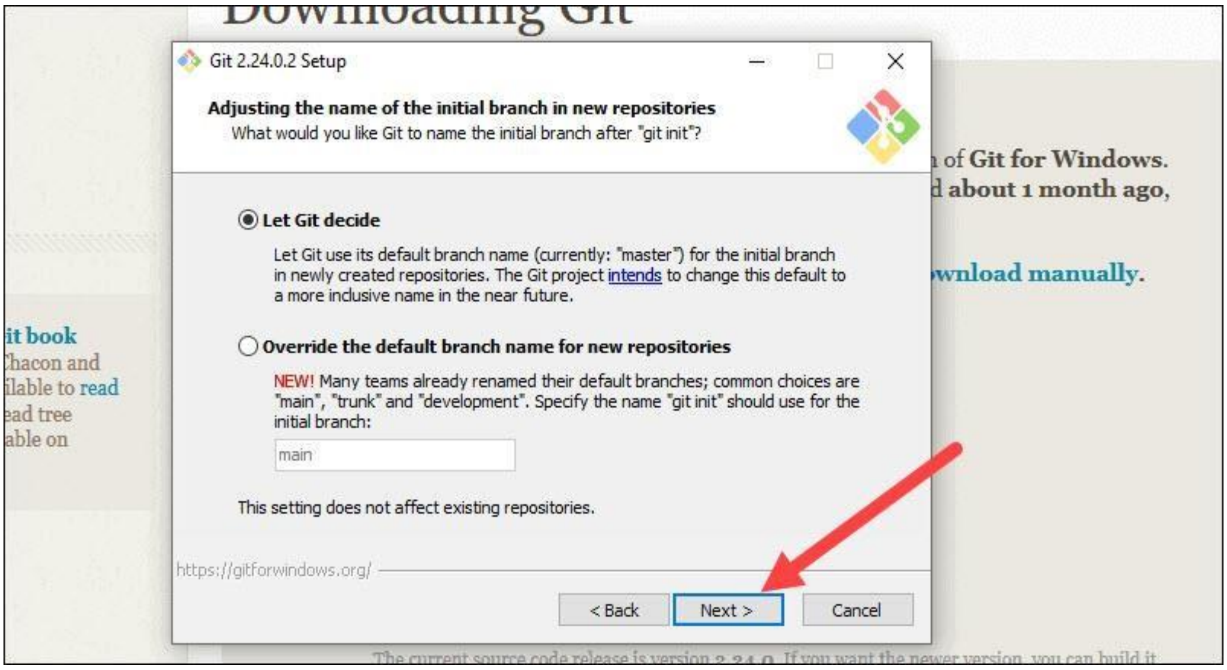
Use Notepad++ as Git's default editor

(NEW!) Notepad++ is a popular GUI editor that can be used by Git.

This editor is popular in part due to the vast number of available plugins; However, when configured via this option, Git will call Notepad++ with plugins disabled (to open the editor as quickly as possible).

https://gitforwindows.org/

The next step allows you to choose a different name for your initial branch. The default is 'master.' Unless you're working in a team that requires a different name, leave the default option and 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**

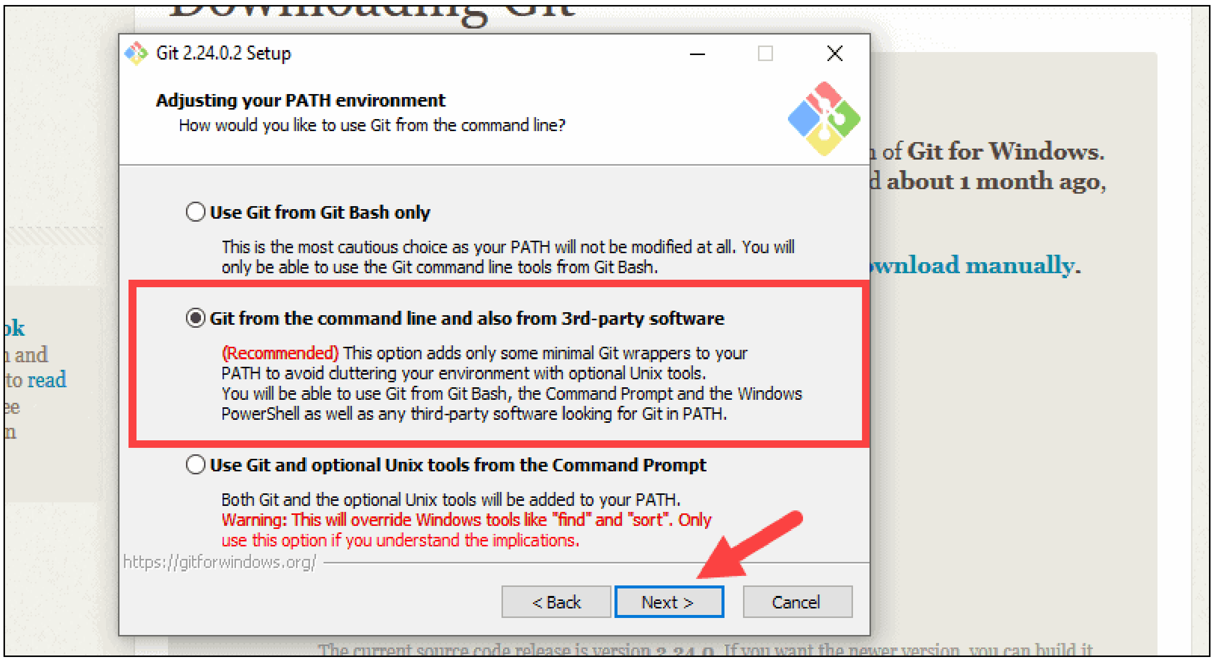
NEW! Many teams already renamed their default branches; common choices are "main", "trunk" and "development". Specify the name "git init" should use for the initial branch:

main

This setting does not affect existing repositories.

https://gitforwindows.org/

This installation step allows you to change the PATH environment. The PATH is the default set of directories included when you run a command from the command line. Leave this on the middle (recommended) selection and click Next.


Git 2.24.0.2 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**

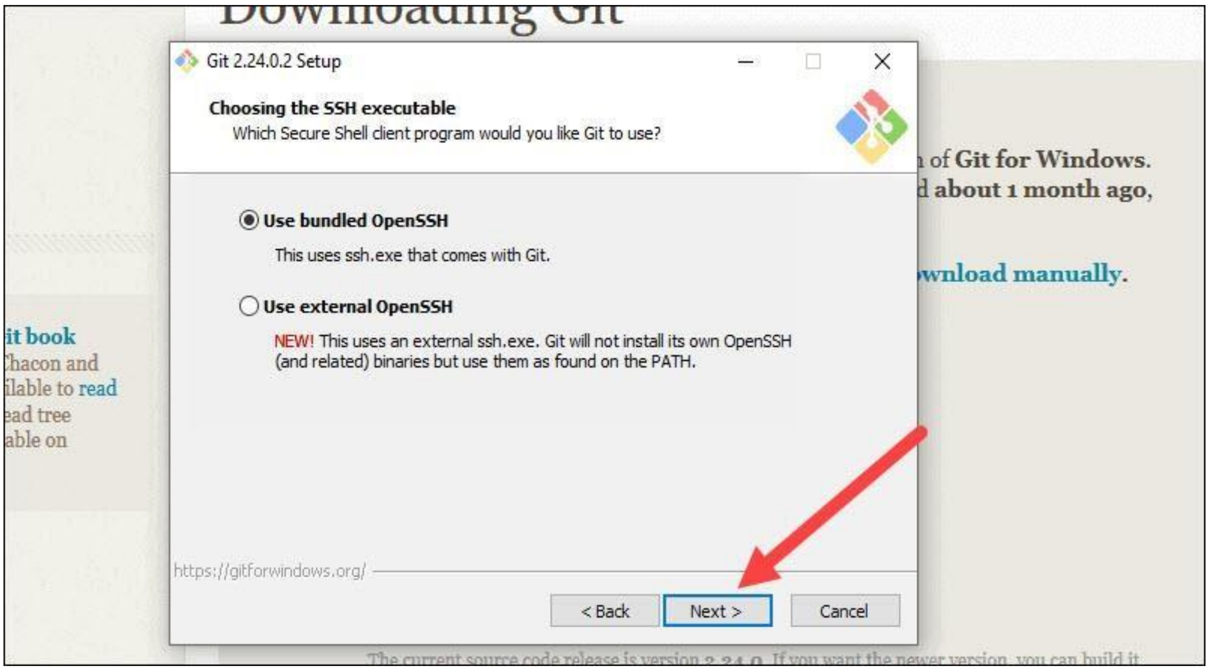
(Recommended) This option adds only some minimal Git wrappers to your PATH to avoid cluttering your environment with optional Unix tools. You will be able to use Git from Git Bash, the Command Prompt and the Windows PowerShell as well as any third-party software looking for Git in PATH.

( ) **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.

Server Certificates, Line Endings and Terminal Emulators

The installer now asks which SSH client you want Git to use. Git already comes with its own SSH client, so if you don't need a specific one, leave the default option and click Next.


Git 2.24.0.2 Setup

Choosing the SSH executable

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

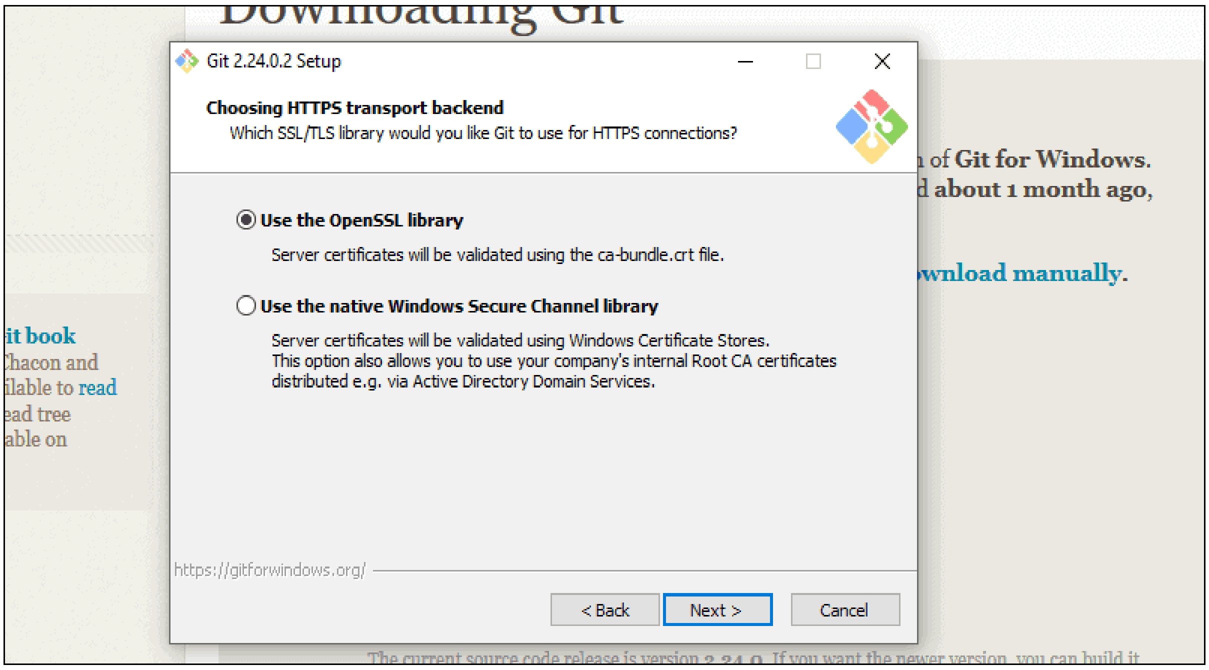
(●) **Use bundled OpenSSH**

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.

The next option relates to server certificates. Most users should use the default. If you're working in an Active Directory environment, you may need to switch to Windows Store certificates. Click Next.


Git 2.24.0.2 Setup

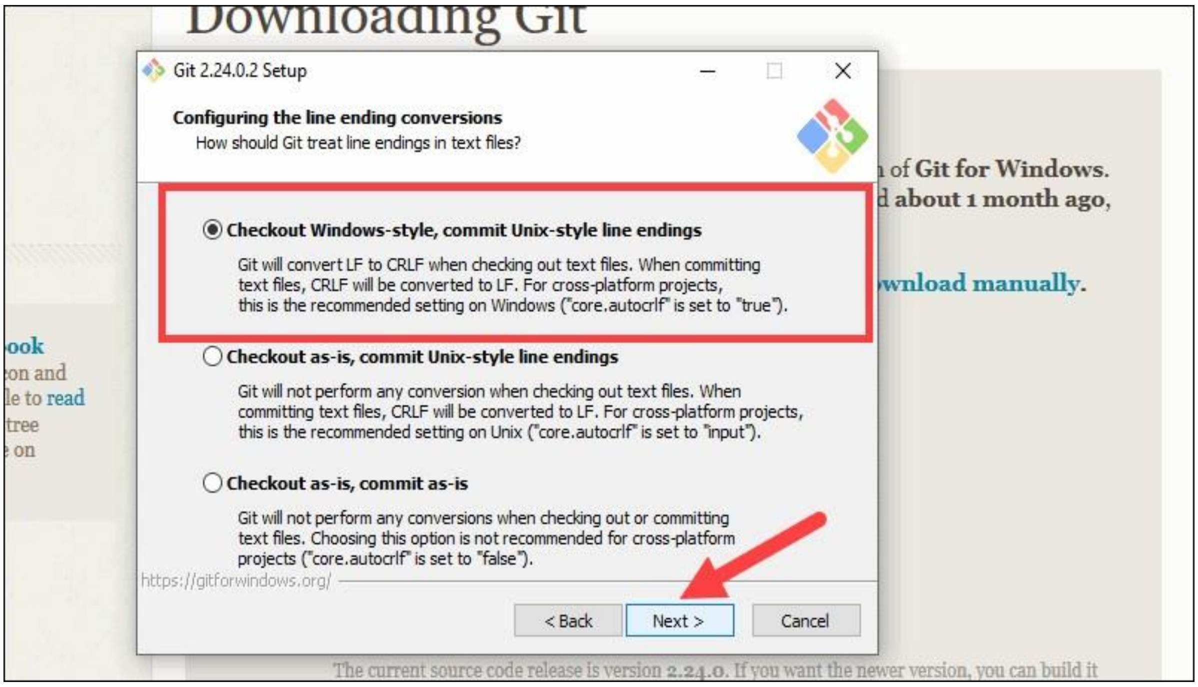
Choosing HTTPS transport backend
Which SSL/TLS library would you like Git to use for HTTPS connections?

[✓ **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/

The next selection converts line endings. It is recommended that you leave the default selection. This relates to the way data is formatted and changing this option may cause problems. Click Next.


Git 2.24.0.2 Setup

Configuring the line ending conversions
How should Git treat line endings in text files?

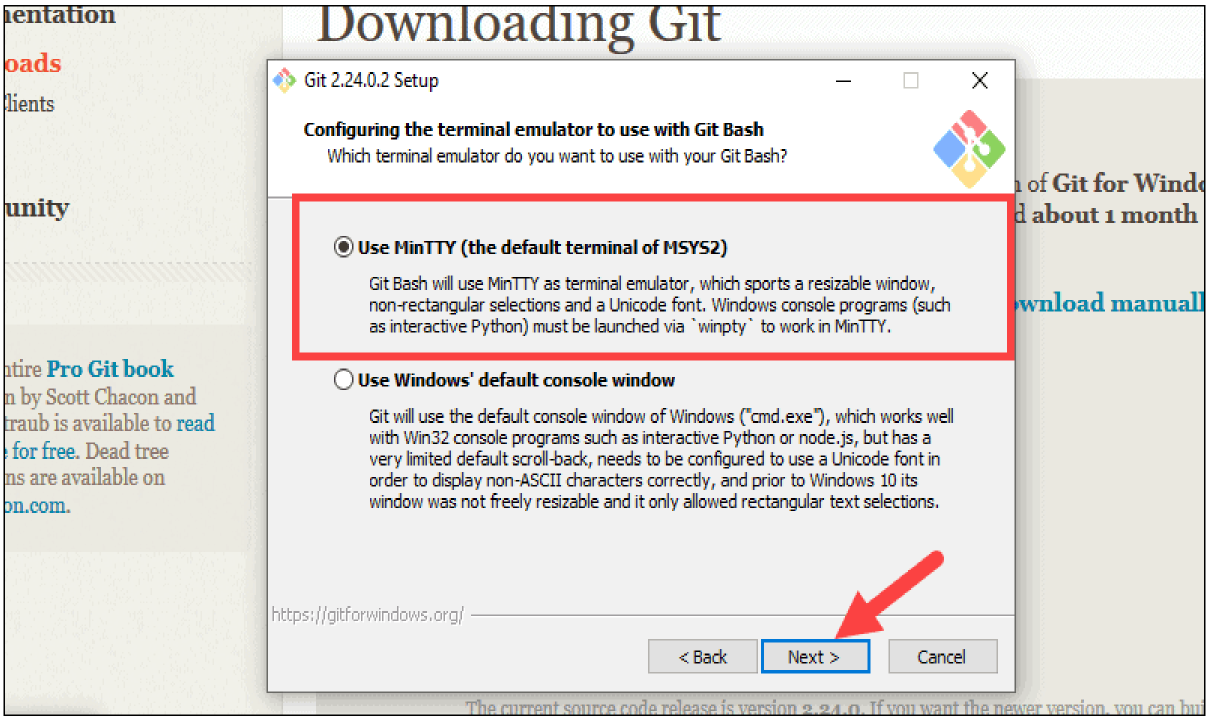
[✓ **Checkout Windows-style, commit Unix-style line endings**]
Git will convert LF to CRLF when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Windows ("core.autocrlf" is set to "true").

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/

Choose the terminal emulator you want to use. The default MinTTY is recommended, for its features. Click Next.



Git 2.24.0.2 Setup

Configuring the terminal emulator to use with Git Bash

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

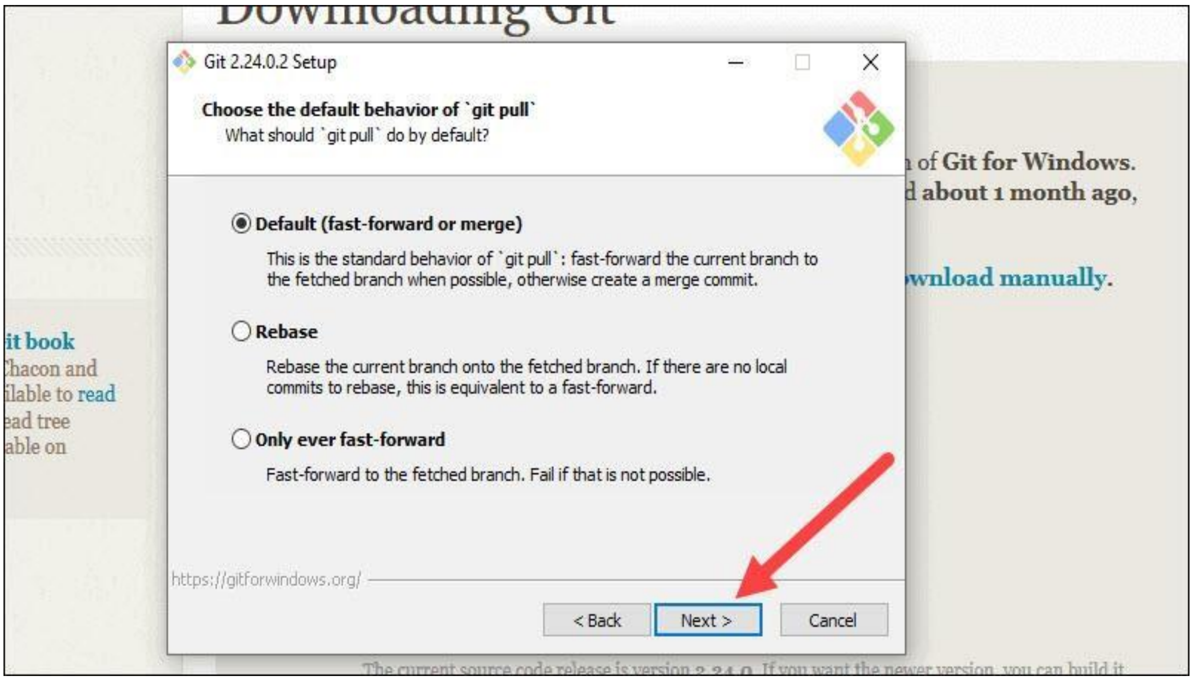
(●) **Use MinTTY (the default terminal of MSYS2)**

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.

The installer now asks what the git pull command should do. The default option is recommended unless you specifically need to change its behaviour. Click Next to continue with the installation.



Git 2.24.0.2 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.

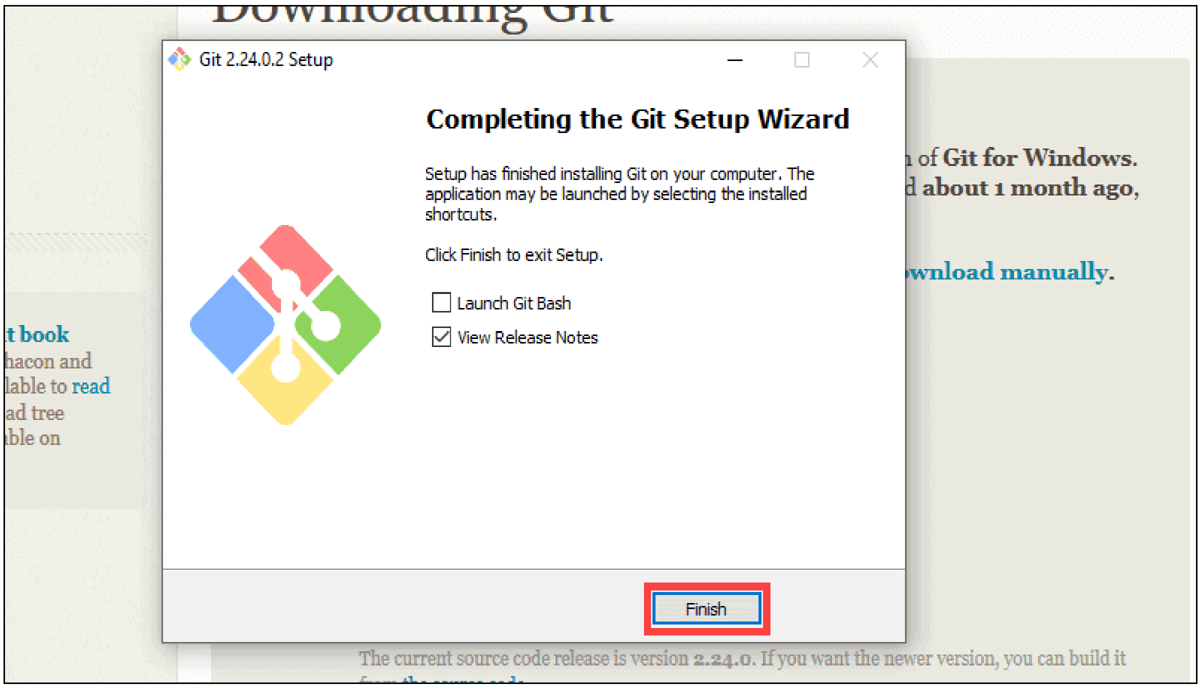
( ) **Rebase**

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.

Once the installation is complete, tick the boxes to view the Release Notes or Launch Git Bash, then click Finish.


Git 2.24.0.2 Setup

Completing the Git Setup Wizard

Setup has finished installing Git on your computer. The application may be launched by selecting the installed shortcuts.

Click Finish to exit Setup.

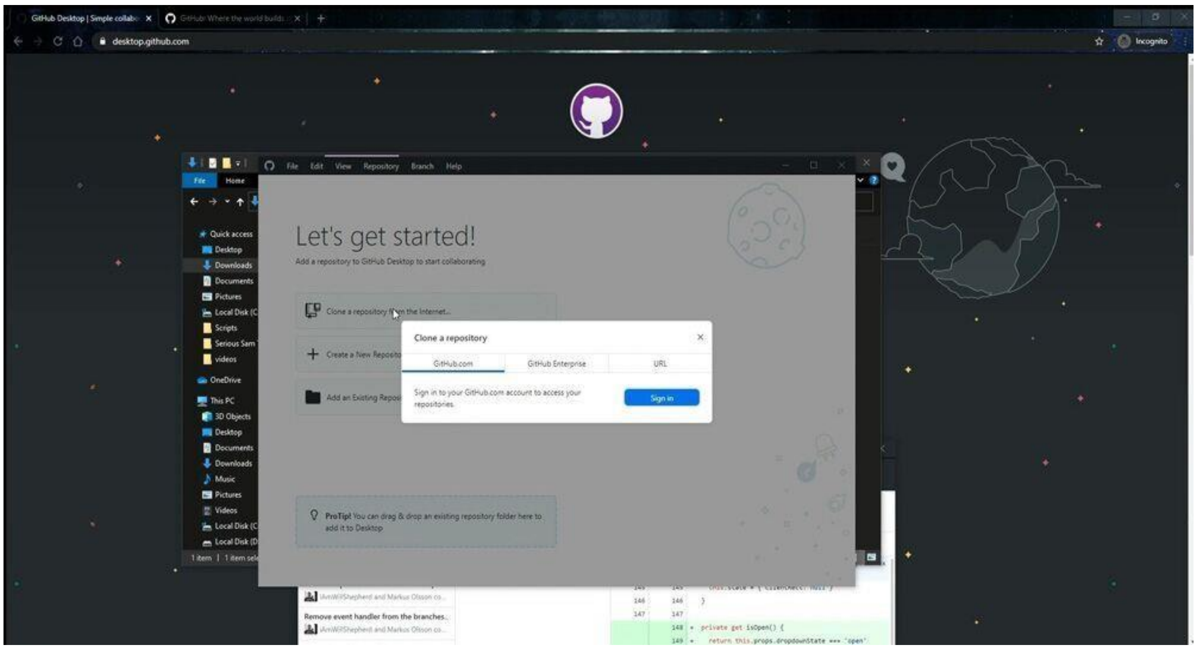
☐ Launch Git Bash

☑ View Release Notes

The current source code release is version 2.24.0. If you want the newer version, you can build it

Installing GITHUB

1.Open Google and search for github desktop. Go to the website with the Github distribution kit. You can find the link in the description. Click on the Download for Windows button. The downloading will start automatically, and the file will be saved in the specified folder. After that, go to the folder with the downloaded file and run it. The installation will be automatic. Now, let's try to clone the repository from Github. In the Github Desktop window, click on the Clone a repository from the Internet.


Let's get started!

Add a repository to GitHub Desktop to start collaborating

Clone a repository from the Internet...

Create a New Repository

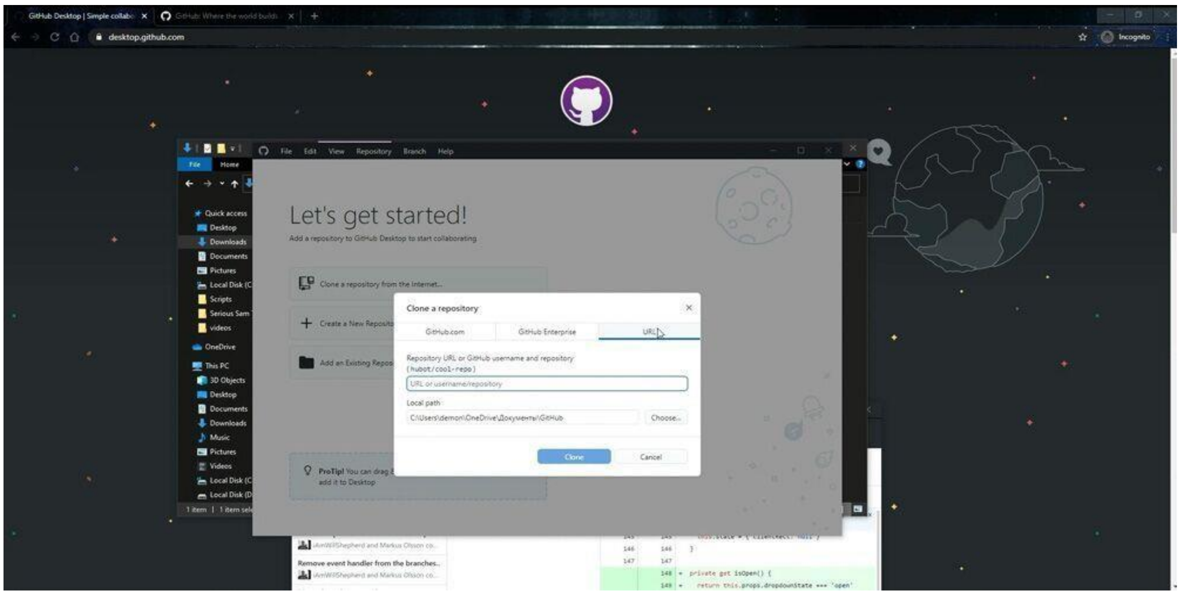
Add an Existing Repository

Clone a repository

Sign in to your GitHub.com account to access your repositories.

ProTip! You can drag & drop an existing repository folder here to add it to Desktop

You can log in to your Github account via the popup window Clone a repository and easily use your own repositories. In case you need to download some other repository, go to the URL tab.


Let's get started!
Add a repository to GitHub Desktop to start collaborating
Clone a repository
GitHub.com
GitHub Enterprise
URL
Repository URL or GitHub username and repository (hubot/cool-repo)
URL or username/repository
Local path
Choose...
Clone
Cancel

Now, you need to paste a link to the repository.

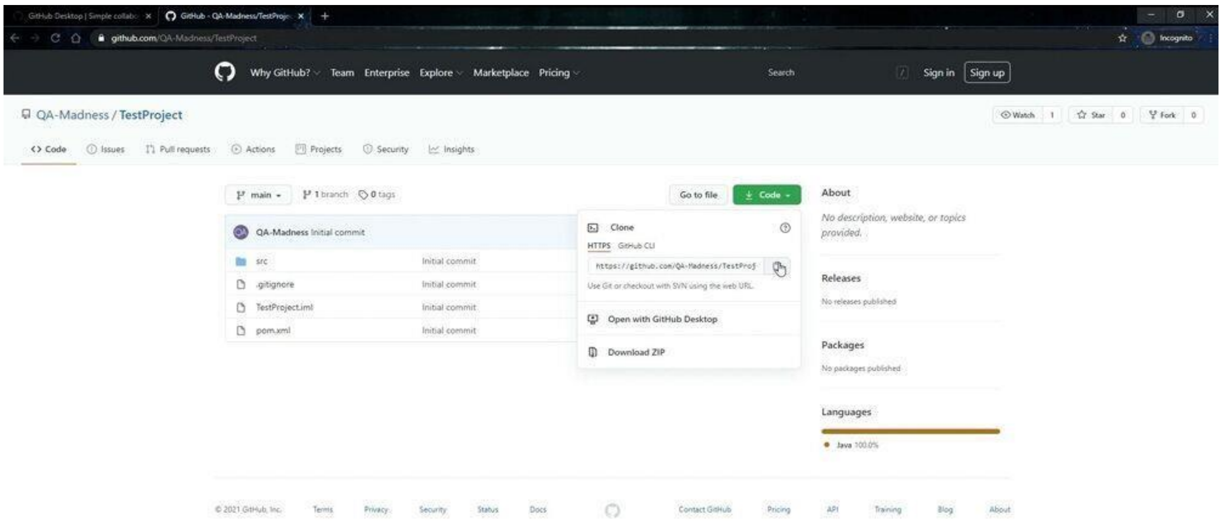
Let’s use our repository as an example. You can find the link to it in the description.

The repository is also available through the search functionality on the GitHub website. Just type QA-Madness in the search bar and press Enter. Select Users in the menu on the results page. You will see QA Madness. Click on it to open the account. You can also subscribe to our page using the Follow button.

Click on the Repositories tab on the account page.

Select TestProject from the list of repositories.

Click on the Code button on the repository page. You will see the link to the repository in the drop-down box. Click to copy it.

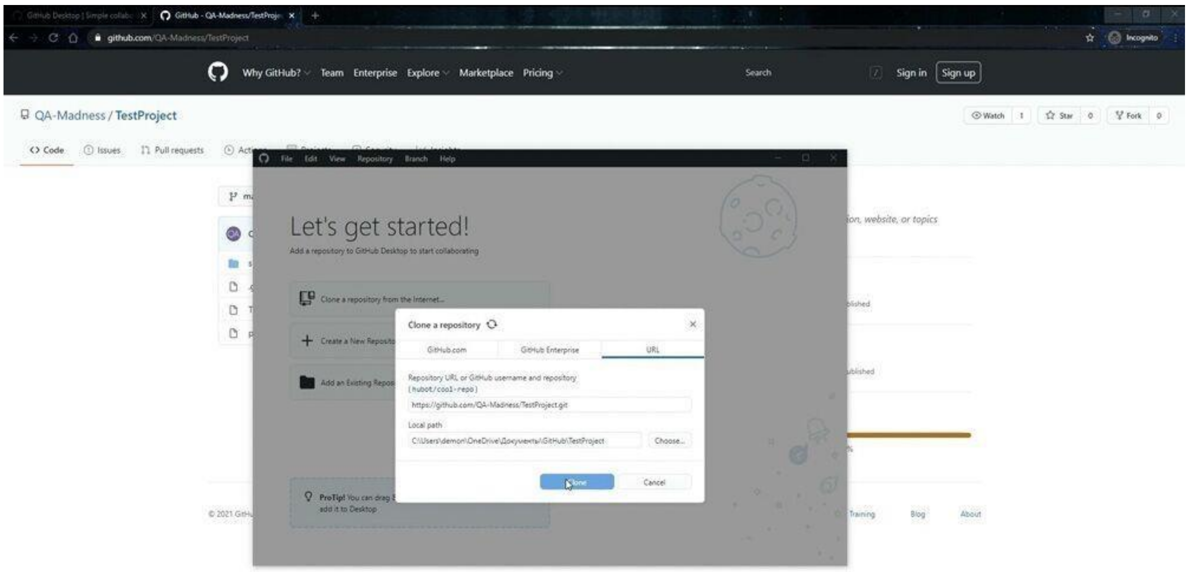


Now, go back to the Github Desktop window.

Paste the copied link in the URL or username/repository bar.

Select the folder for the download.

Click on the Clone.


Let's get started!

Add a repository to GitHub Desktop to start collaborating

Clone a repository

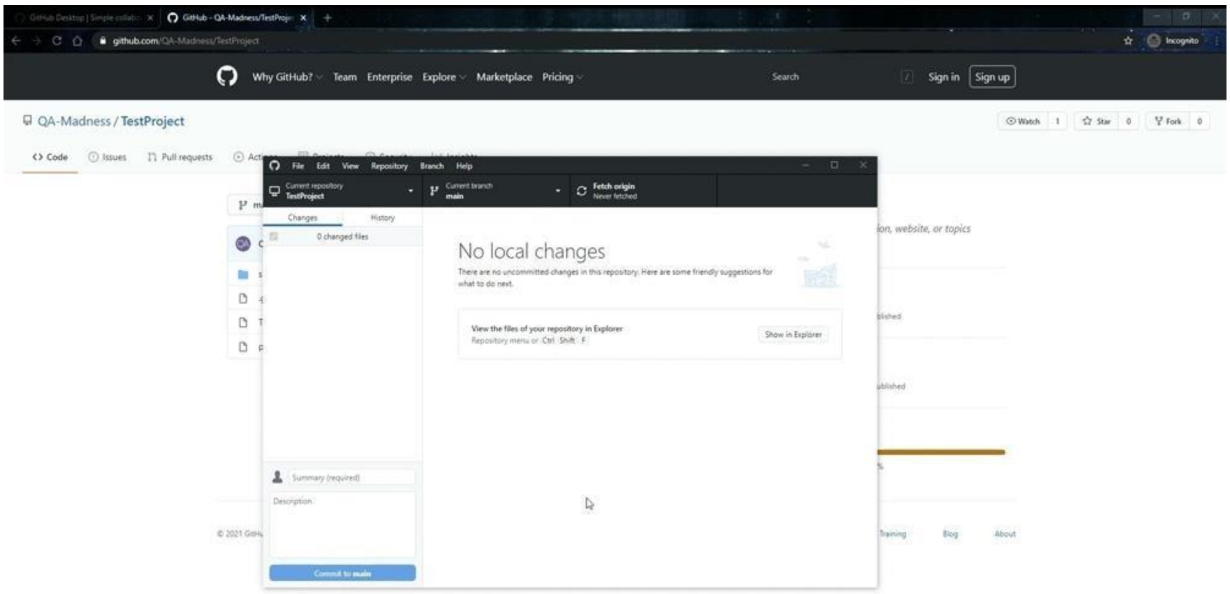
Repository URL or GitHub username and repository (hubot/cool-repo)

https://github.com/QA-Madness/TestProject.git

Local path

C:\Users\demon\OneDrive\Документы\GitHub\TestProject

After this, you can then open the folder with the downloaded repository and make sure it is there.


Current repository: TestProject

Current branch: main

Fetch origin: Never fetched

0 changed files

No local changes

There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.

View the files of your repository in Explorer

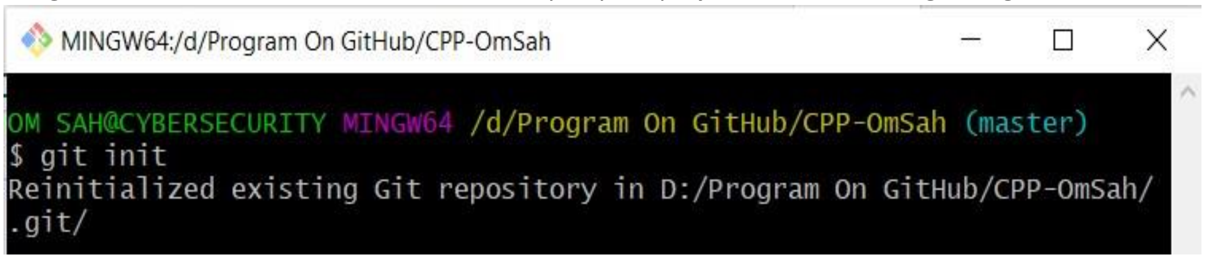
Repository menu or Ctrl Shift F

Result : The Software GIT and GITHUB has been installed successfully.

Exp. No : 2. Basic Commands on GIT (GIT cheat sheet)

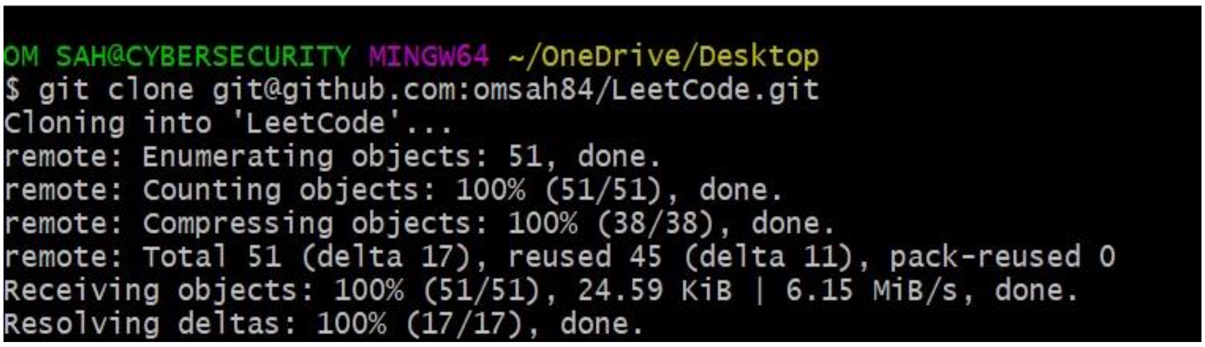
Aim: To Learn and execute the various Basic commands on GIT cheat sheet.

Command: git init
Usage: Run this command in the root directory of your project to start tracking changes.



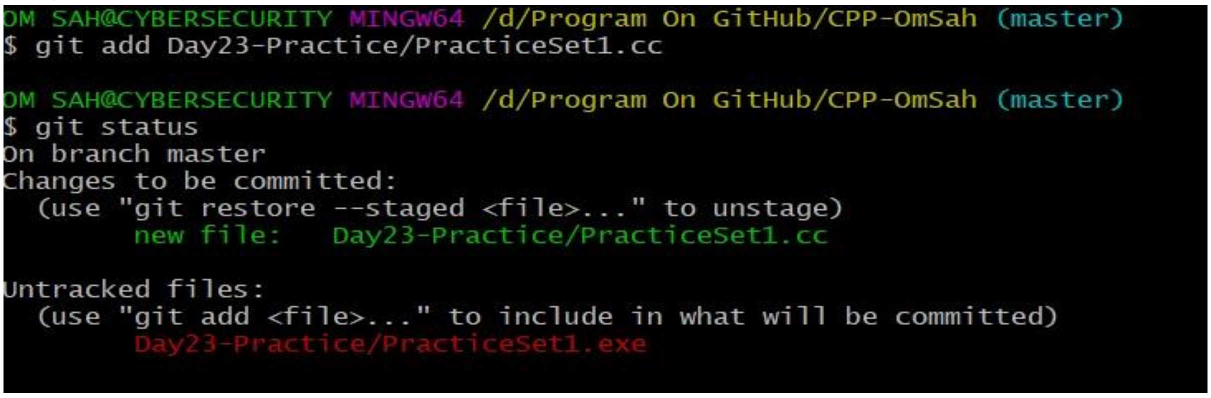
```
MINGW64:/d/Program On GitHub/CPP-OmSah

OM SAH@CYBERSECURITY MINGW64 /d/Program On GitHub/CPP-OmSah (master)
$ git init
Reinitialized existing Git repository in D:/Program On GitHub/CPP-OmSah/
.git/
```

Command: git clone <repository-url>
Usage: Use this command to create a copy of a remote repository on your local machine.



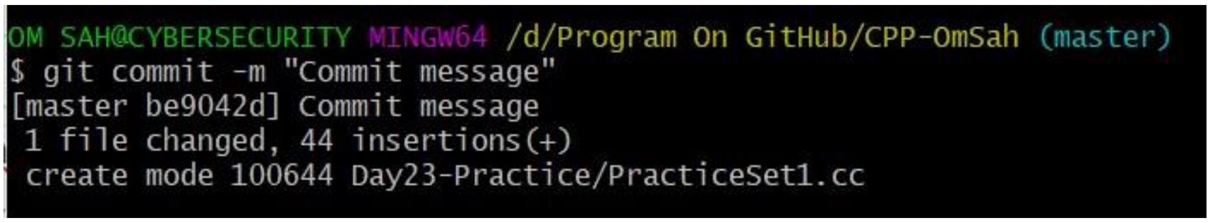
```
OM SAH@CYBERSECURITY MINGW64 ~/OneDrive/Desktop
$ git clone git@github.com:omsah84/LeetCode.git
Cloning into 'LeetCode'...
remote: Enumerating objects: 51, done.
remote: Counting objects: 100% (51/51), done.
remote: Compressing objects: 100% (38/38), done.
remote: Total 51 (delta 17), reused 45 (delta 11), pack-reused 0
Receiving objects: 100% (51/51), 24.59 KiB | 6.15 MiB/s, done.
Resolving deltas: 100% (17/17), done.
```

Command: git add <file> or git add . (to add all changes)
Usage: Select specific files or all files to be included in the next commit.



```
OM SAH@CYBERSECURITY MINGW64 /d/Program On GitHub/CPP-OmSah (master)
$ git add Day23-Practice/PracticeSet1.cc

OM SAH@CYBERSECURITY MINGW64 /d/Program On GitHub/CPP-OmSah (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   Day23-Practice/PracticeSet1.cc

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Day23-Practice/PracticeSet1.exe
```

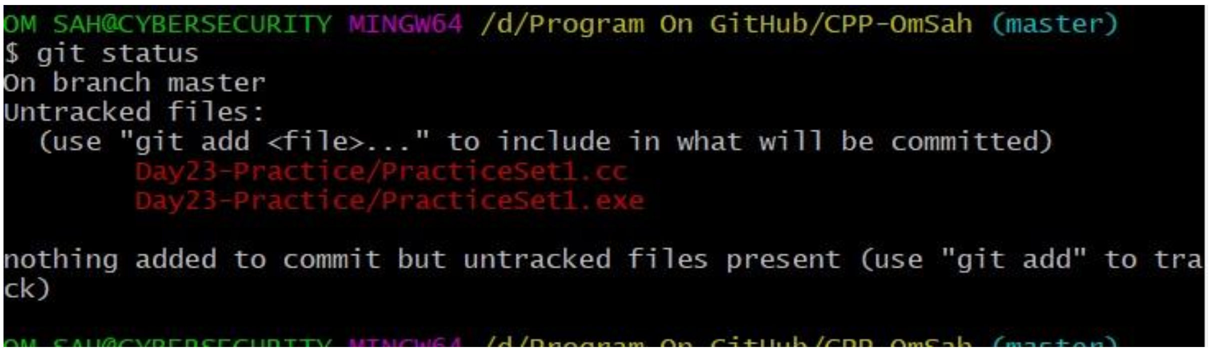
Command: git commit -m "Your commit message"
Usage: Save staged changes with a descriptive commit message.



```
OM SAH@CYBERSECURITY MINGW64 /d/Program On GitHub/CPP-OmSah (master)
$ git commit -m "Commit message"
[master be9042d] Commit message
 1 file changed, 44 insertions(+)
 create mode 100644 Day23-Practice/PracticeSet1.cc
```

Command: git status

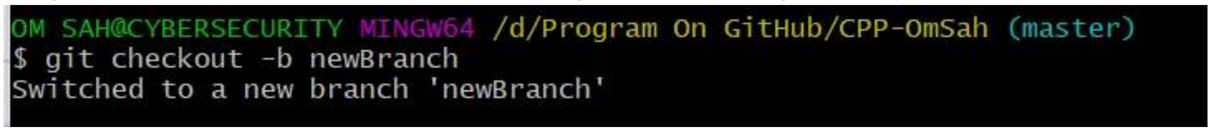
Usage: Check the current state of your working directory and staging area.



```
OM SAH@CYBERSECURITY MINGW64 /d/Program On GitHub/CPP-OmSah (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Day23-Practice/PracticeSet1.cc
        Day23-Practice/PracticeSet1.exe

nothing added to commit but untracked files present (use "git add" to tra
ck)
```

Command: git checkout <branch-name> or git checkout -- <file>

Usage: Move between branches or discard changes in the working directory.



```
OM SAH@CYBERSECURITY MINGW64 /d/Program On GitHub/CPP-OmSah (master)
$ git checkout -b newBranch
Switched to a new branch 'newBranch'
```

Result : The various GIT cheat sheet commands has been executed successfully.

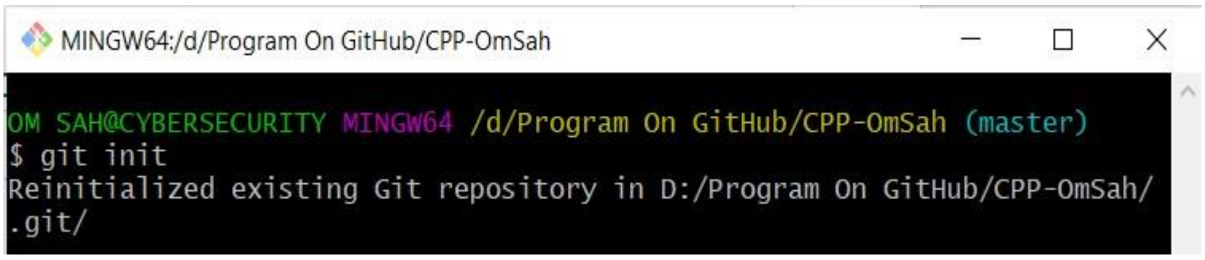
Exp. No : 3. Basic Commands on GITHUB (GITHUB Cheat sheet)

Aim: To Learn basic commands on GitHub for version control and collaboration.

Procedure:

Creating a Repository:

Command: git init

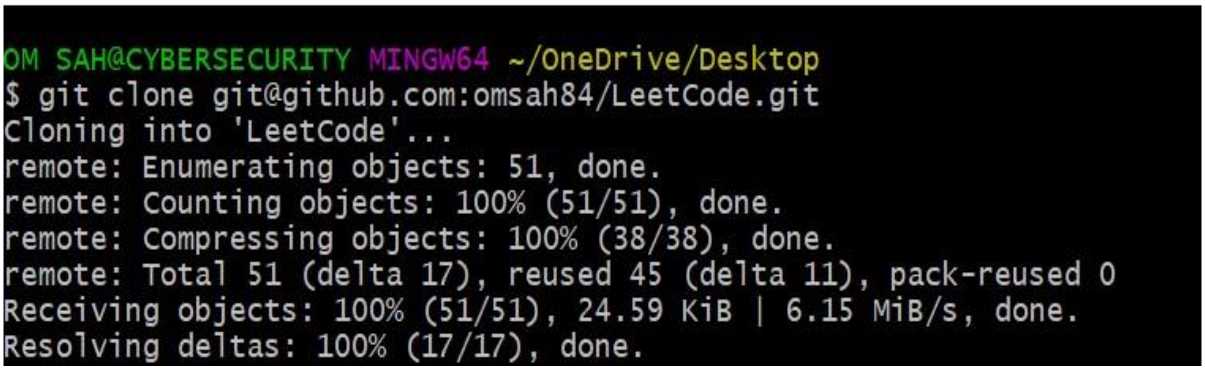


```
MINGW64:/d/Program On GitHub/CPP-OmSah
OM SAH@CYBERSECURITY MINGW64 /d/Program On GitHub/CPP-OmSah (master)
$ git init
Reinitialized existing Git repository in D:/Program On GitHub/CPP-OmSah/
.git/
```

Result: Initializes a new Git repository in the current directory.

Cloning a Repository:

Command: git clone <repository_url>

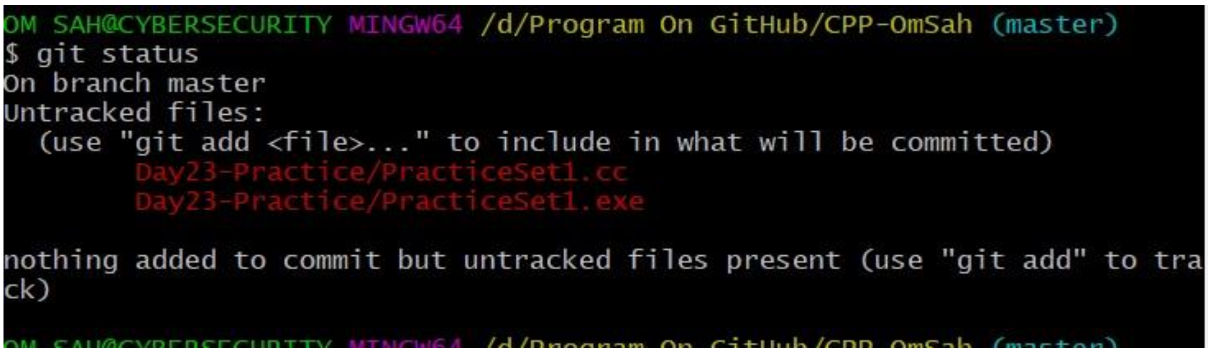


```
OM SAH@CYBERSECURITY MINGW64 ~/OneDrive/Desktop
$ git clone git@github.com:omsah84/LeetCode.git
Cloning into 'LeetCode'...
remote: Enumerating objects: 51, done.
remote: Counting objects: 100% (51/51), done.
remote: Compressing objects: 100% (38/38), done.
remote: Total 51 (delta 17), reused 45 (delta 11), pack-reused 0
Receiving objects: 100% (51/51), 24.59 KiB | 6.15 MiB/s, done.
Resolving deltas: 100% (17/17), done.
```

Result: Creates a copy of a remote repository on your local machine.

Checking Repository Status:

Command: git status

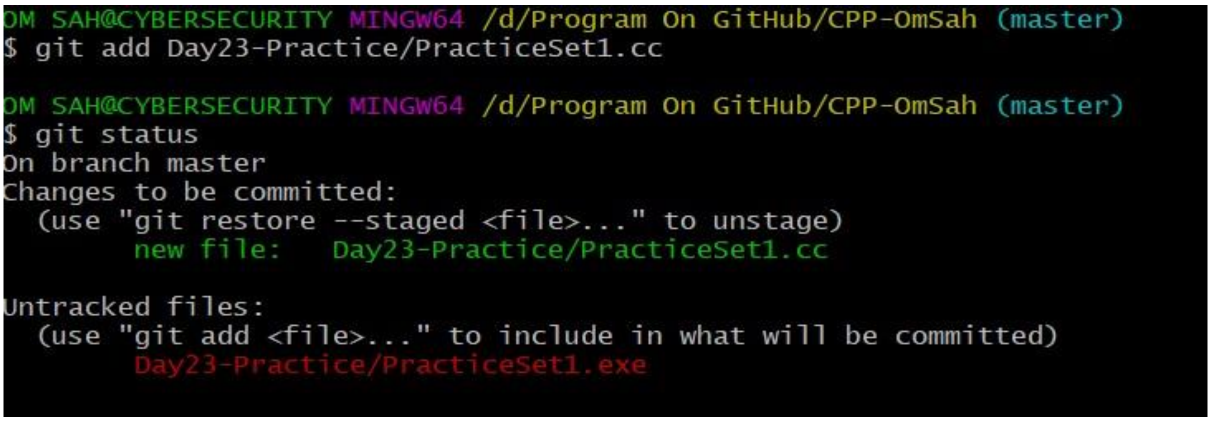


```
OM SAH@CYBERSECURITY MINGW64 /d/Program On GitHub/CPP-OmSah (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Day23-Practice/PracticeSet1.cc
        Day23-Practice/PracticeSet1.exe

nothing added to commit but untracked files present (use "git add" to tra
ck)
```

Result: Shows the status of changes as untracked, modified, or staged.

Adding Changes to Staging Area:

Command: git add <file_name>

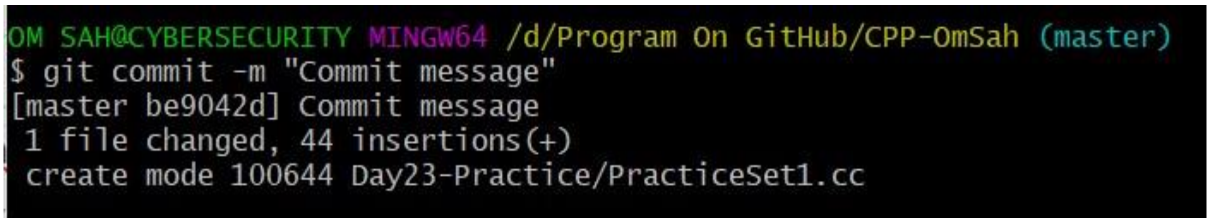


```
OM SAH@CYBERSECURITY MINGW64 /d/Program On GitHub/CPP-OmSah (master)
$ git add Day23-Practice/PracticeSet1.cc

OM SAH@CYBERSECURITY MINGW64 /d/Program On GitHub/CPP-OmSah (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   Day23-Practice/PracticeSet1.cc

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Day23-Practice/PracticeSet1.exe
```

Result: Stages changes, ready for commit.

Committing Changes:

Command: git commit -m "Commit message"

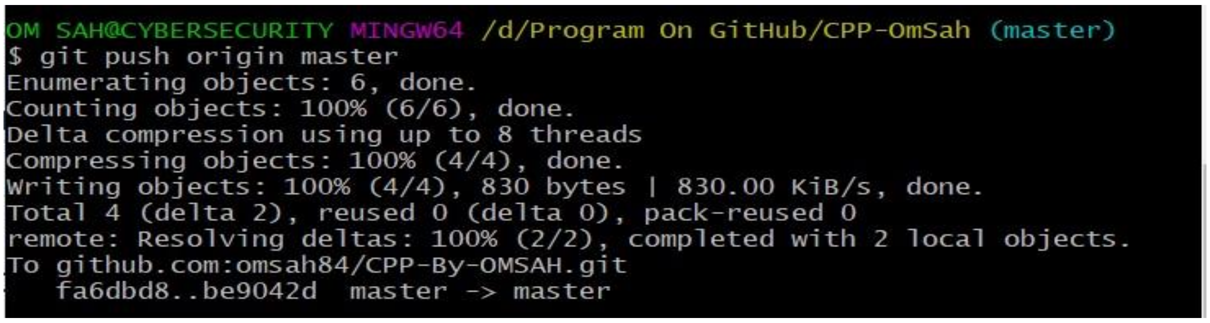


```
OM SAH@CYBERSECURITY MINGW64 /d/Program On GitHub/CPP-OmSah (master)
$ git commit -m "Commit message"
[master be9042d] Commit message
 1 file changed, 44 insertions(+)
 create mode 100644 Day23-Practice/PracticeSet1.cc
```

Result: Records changes to the repository along with a descriptive message.

Pushing Changes to Remote Repository:

Command: git push origin <branch_name>

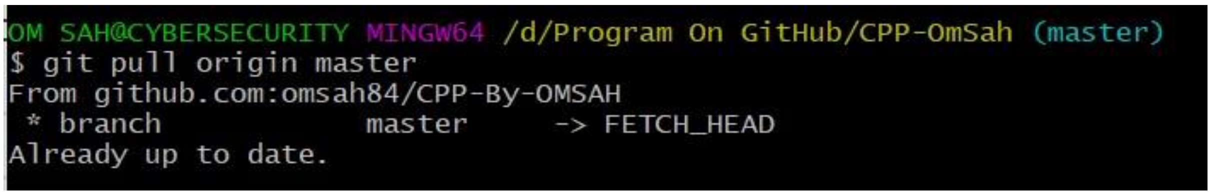


```
OM SAH@CYBERSECURITY MINGW64 /d/Program On GitHub/CPP-OmSah (master)
$ git push origin master
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 830 bytes | 830.00 KiB/s, done.
Total 4 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:omsah84/CPP-By-OMSAH.git
   fa6dbd8..be9042d  master -> master
```

Result: Uploads local branch commits to the remote repository.

Pulling Changes from Remote Repository:

Command: git pull origin <branch_name>

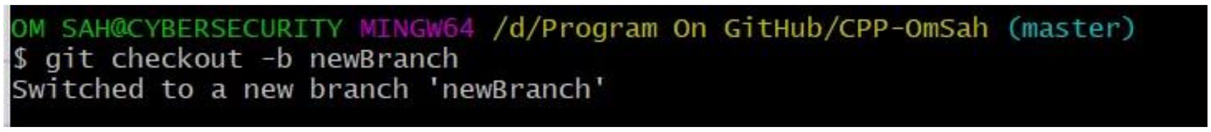


```
OM SAH@CYBERSECURITY MINGW64 /d/Program On GitHub/CPP-OmSah (master)
$ git pull origin master
From github.com:omsah84/CPP-By-OMSAH
 * branch            master     -> FETCH_HEAD
Already up to date.
```

Result: Fetches changes from a remote repository and merges them into the current branch.

Creating a New Branch:

Command: git checkout -b <new_branch_name>

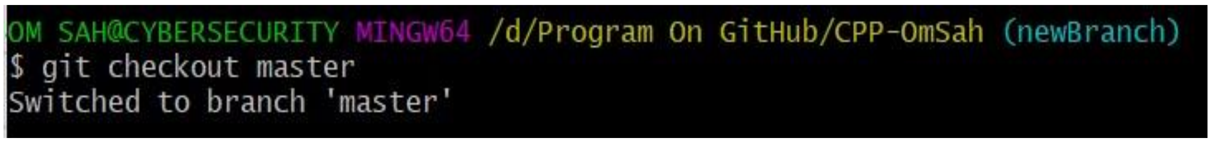


```
OM SAH@CYBERSECURITY MINGW64 /d/Program On GitHub/CPP-OmSah (master)
$ git checkout -b newBranch
Switched to a new branch 'newBranch'
```

Result: Creates a new branch and switches to it.

Switching Branches:

Command: git checkout <branch_name>

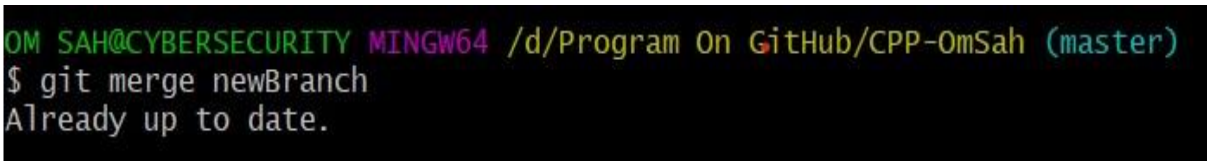


```
OM SAH@CYBERSECURITY MINGW64 /d/Program On GitHub/CPP-OmSah (newBranch)
$ git checkout master
Switched to branch 'master'
```

Result: Switches to an existing branch.

Merging Branches:

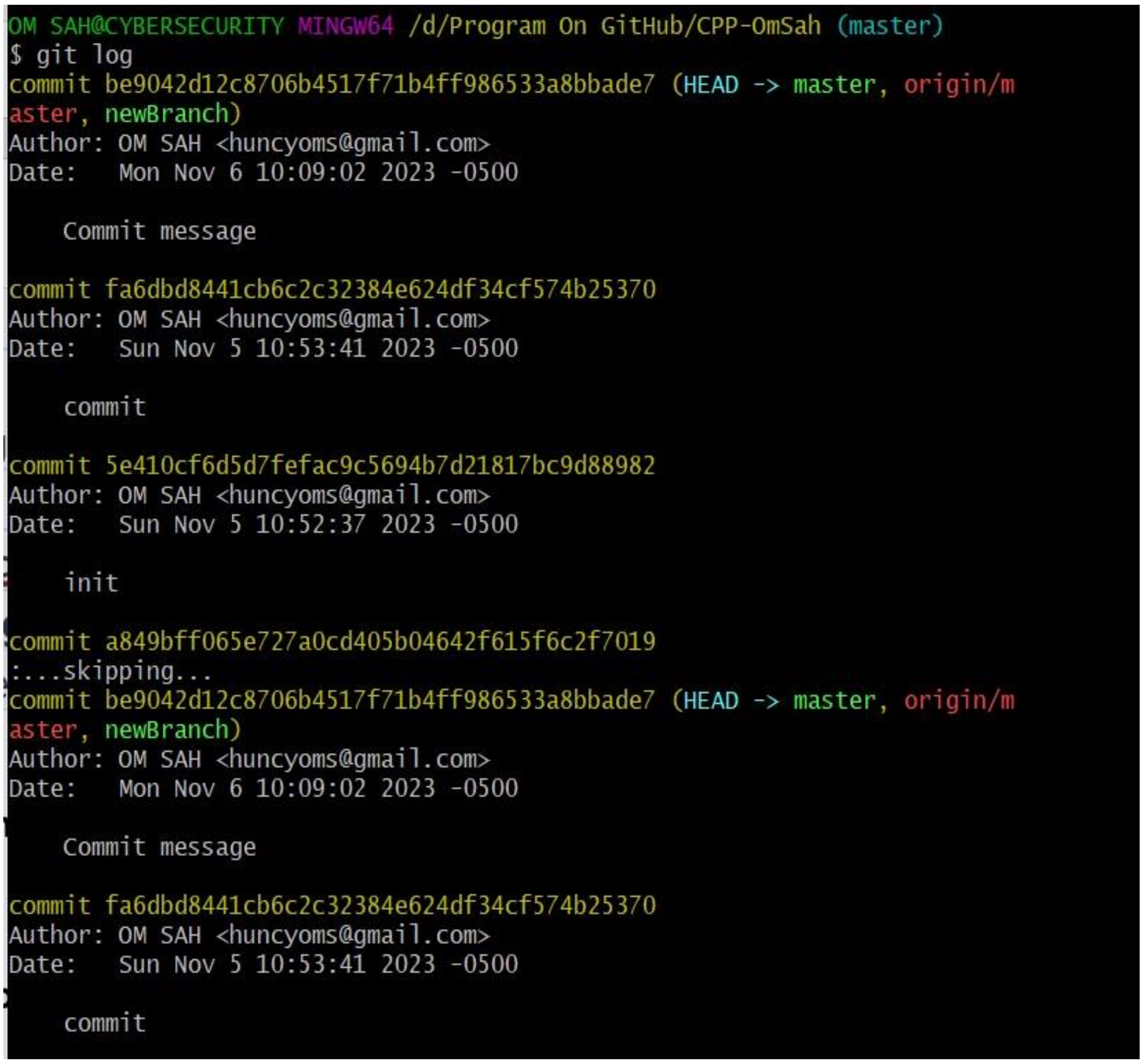
Command: git merge <branch_name>



```
OM SAH@CYBERSECURITY MINGW64 /d/Program On GitHub/CPP-OmSah (master)
$ git merge newBranch
Already up to date.
```

Result: Combines changes from one branch into another.

Viewing Commit History:
Command: git log



```
OM SAH@CYBERSECURITY MINGW64 /d/Program On GitHub/CPP-OmSah (master)
$ git log
commit be9042d12c8706b4517f71b4ff986533a8bbade7 (HEAD -> master, origin/m
aster, newBranch)
Author: OM SAH <huncyoms@gmail.com>
Date:   Mon Nov 6 10:09:02 2023 -0500

    Commit message

commit fa6dbd8441cb6c2c32384e624df34cf574b25370
Author: OM SAH <huncyoms@gmail.com>
Date:   Sun Nov 5 10:53:41 2023 -0500

    commit

commit 5e410cf6d5d7fefac9c5694b7d21817bc9d88982
Author: OM SAH <huncyoms@gmail.com>
Date:   Sun Nov 5 10:52:37 2023 -0500

    init

commit a849bff065e727a0cd405b04642f615f6c2f7019
:...skipping...
commit be9042d12c8706b4517f71b4ff986533a8bbade7 (HEAD -> master, origin/m
aster, newBranch)
Author: OM SAH <huncyoms@gmail.com>
Date:   Mon Nov 6 10:09:02 2023 -0500

    Commit message

commit fa6dbd8441cb6c2c32384e624df34cf574b25370
Author: OM SAH <huncyoms@gmail.com>
Date:   Sun Nov 5 10:53:41 2023 -0500

    commit
```

Result: Displays a log of all commits in the current branch.

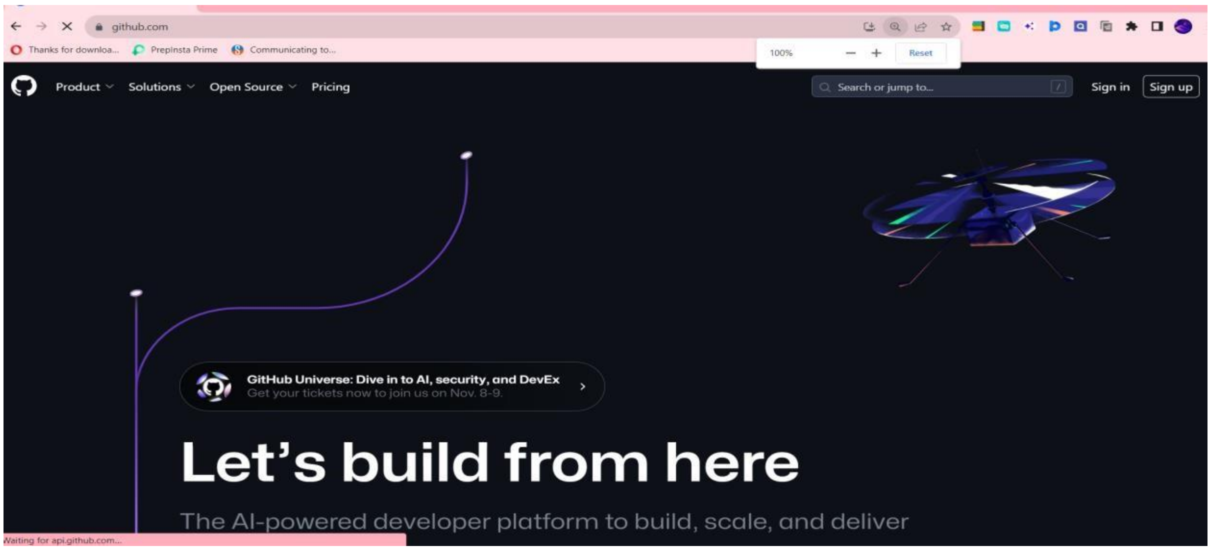
Result : Various commands on GitHub for version control and collaboration has been executed successfully.

Exp. No : 4. Create a "repository" (project) with a git hosting tool.

Aim: Create a repository (project) on a GitHub to facilitate collaborative version control and project management.

Procedure:

Sign in or Create an Account: If you don’t have a GitHub account, you’ll need to sign up. If you already have an account, sign in to your GitHub account.

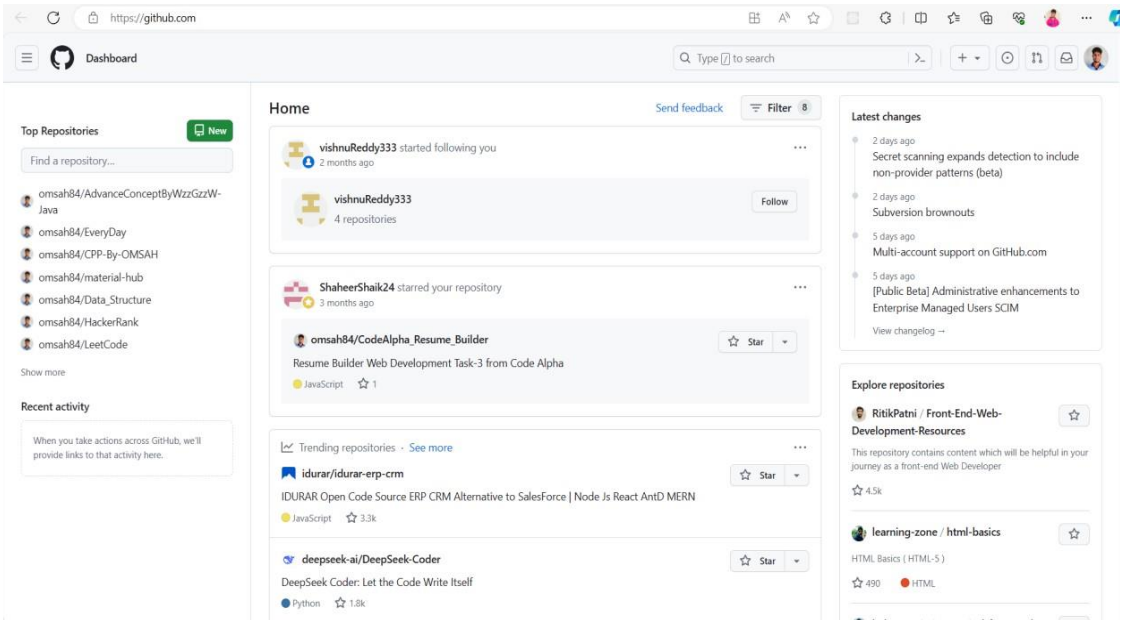


GitHub Universe: Dive in to AI, security, and DevEx
Get your tickets now to join us on Nov. 8-9.

Let’s build from here

The AI-powered developer platform to build, scale, and deliver

Create a New Repository:
On the top right corner of the GitHub homepage, you will find a ”+” icon. Click on it, and from the dropdown menu, select “New repository”.



Top Repositories
omsah84/AdvanceConceptByWzzGzzW-Java
omsah84/EveryDay
omsah84/CPP-By-OMSAH
omsah84/material-hub
omsah84/Data_Structure
omsah84/HackerRank
omsah84/LeetCode

Recent activity
When you take actions across GitHub, we'll provide links to that activity here.

Home

vishnuReddy333 started following you
2 months ago
vishnuReddy333
4 repositories

ShaheerShaik24 starred your repository
3 months ago
omsah84/CodeAlpha_Resume_Builder
Resume Builder Web Development Task-3 from Code Alpha
JavaScript 1

Trending repositories
idurar/idurar-erp-crm
IDURAR Open Code Source ERP CRM Alternative to SalesForce | Node Js React AntD MERN
JavaScript 3.3k

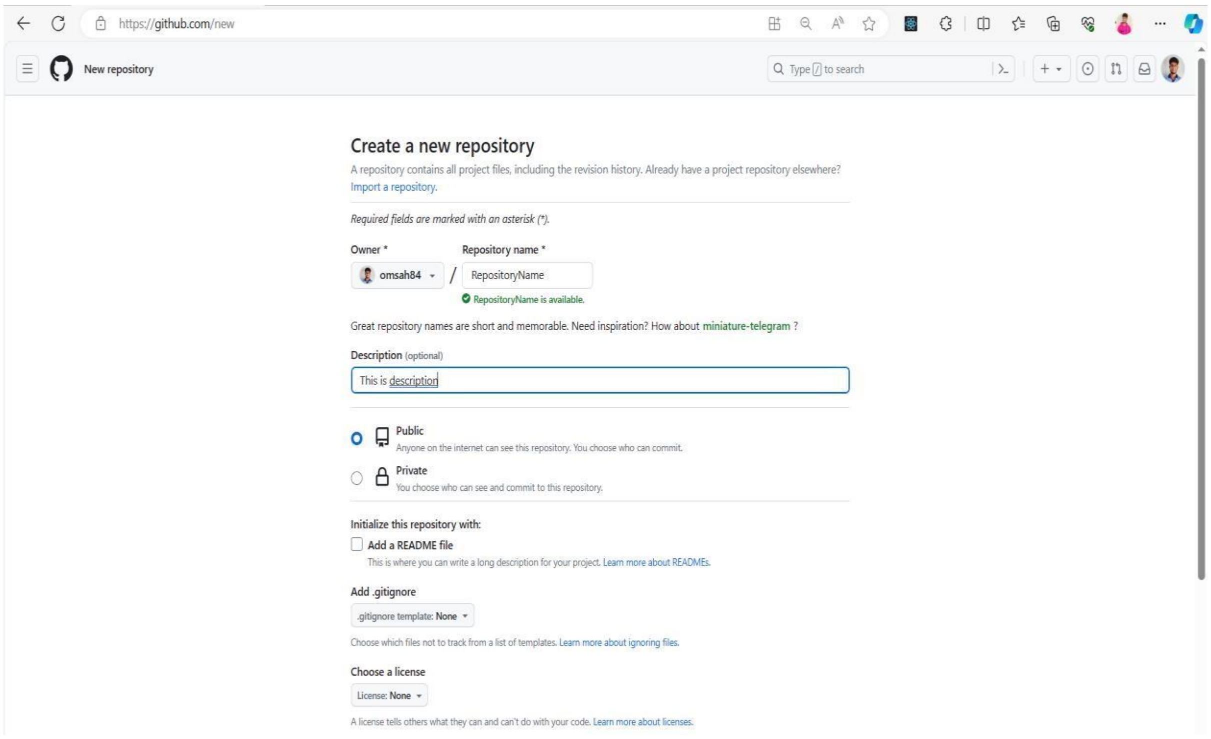
deepseek-ai/DeepSeek-Coder
DeepSeek Coder: Let the Code Write Itself
Python 1.8k

Latest changes
2 days ago
Secret scanning expands detection to include non-provider patterns (beta)
2 days ago
Subversion brownouts
5 days ago
Multi-account support on GitHub.com
5 days ago
[Public Beta] Administrative enhancements to Enterprise Managed Users SCIM

Explore repositories
RitikPatni / Front-End-Web-Development-Resources
This repository contains content which will be helpful in your journey as a front-end Web Developer
4.5k
learning-zone / html-basics
HTML Basics (HTML-5)
490 HTML

Fill out the Repository Details:
Give your repository a name.
Add a description to explain what your project is about (optional) Choose the visibility of your repository (public or private).

Initialize this repository with a README if you want to create a README file for your project. It’s a good practice to have a README file explaining your project.



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 * omsah84 / Repository name * RepositoryName

RepositoryName is available.

Great repository names are short and memorable. Need inspiration? How about miniature-telegram ?

Description (optional)

This is description

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.

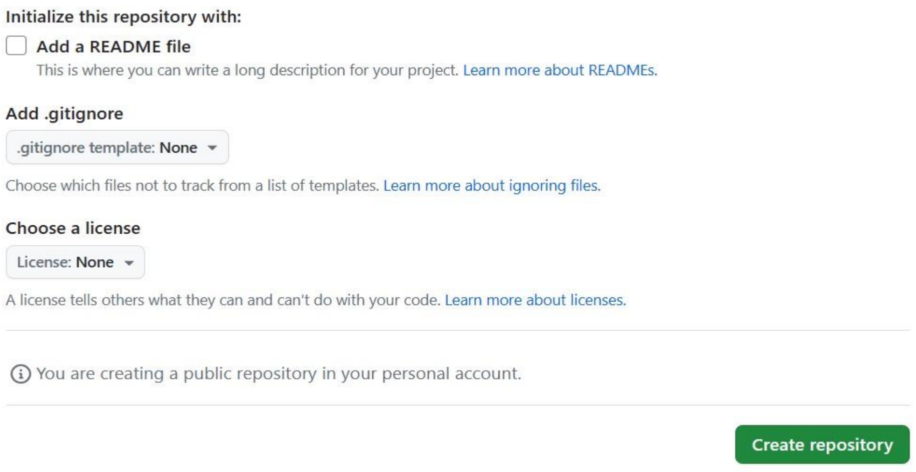
Choose a License and .gitignore File (Optional):

You cand choose license for your project. GitHub provides several open-source licenses you can select from.

You can also add a .gitignore file. This file specifies intentionally untracked files that Git should ignore.

Create Repository:

Click on the “Create repository” button to create your 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.

You are creating a public repository in your personal account.

Result : Created a repository for a project on a GitHub to facilitate collaborative version control and project management that has been executed successfully.

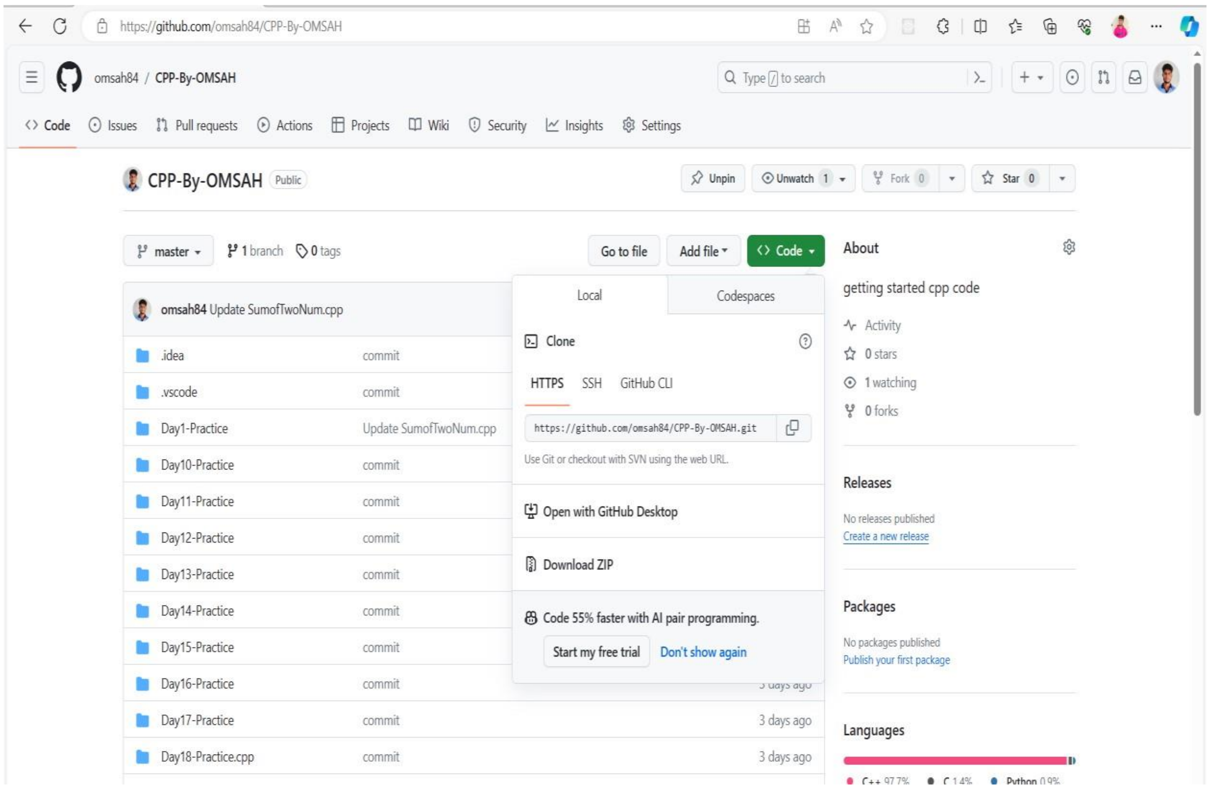
Exp. No : 5. Copy (or clone) the repository to your local machine

Aim: Creating a Copy (or Clone) the Repository to Your Local Machine.

Definition: The aim of this step is to create a local copy of the repository that exists on the Git hosting platform (such as GitHub) onto your personal computer. This local copy allows you to work on the project, make changes, and collaborate without directly modifying the original repository on the remote server.

Clone the Repository:

Definition: Cloning a repository means creating an identical copy of the remote repository on your local machine. It not only copies the files but also sets up the necessary Git configuration, such as remote tracking branches, enabling you to work seamlessly with the repository locally.



omsah84 Update SumofTwoNum.cpp

.idea	commit	
.vscode	commit	
Day1-Practice	Update SumofTwoNum.cpp	
Day10-Practice	commit	
Day11-Practice	commit	
Day12-Practice	commit	
Day13-Practice	commit	
Day14-Practice	commit	
Day15-Practice	commit	
Day16-Practice	commit	3 days ago
Day17-Practice	commit	3 days ago
Day18-Practice.cpp	commit	3 days ago

https://github.com/omsah84/CPP-By-OMSAH.git

Use Git or checkout with SVN using the web URL.

Code 55% faster with AI pair programming.

getting started cpp code

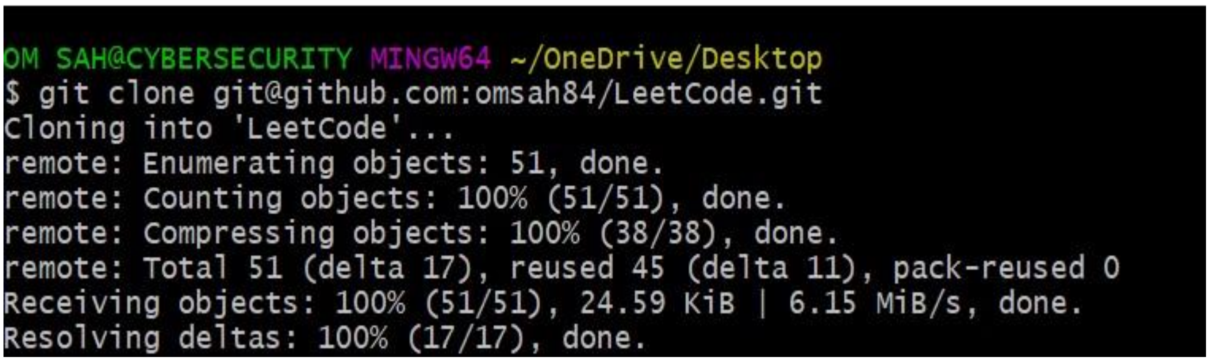
No releases published
Create a new release

No packages published
Publish your first package

C++ 97.7% C 1.4% Python 0.9%

How to Clone:

Use the following Git command in your terminal or command prompt:



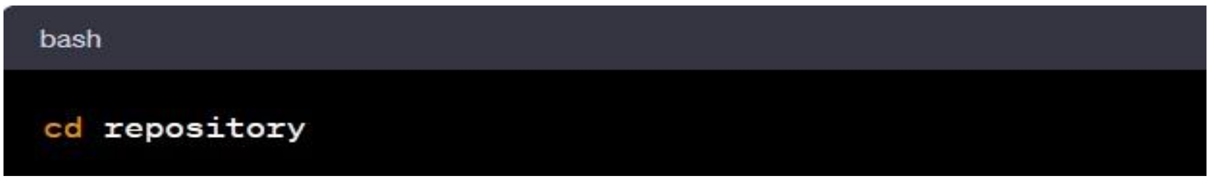
```
OM SAH@CYBERSECURITY MINGW64 ~/OneDrive/Desktop
$ git clone git@github.com:omsah84/LeetCode.git
Cloning into 'LeetCode'...
remote: Enumerating objects: 51, done.
remote: Counting objects: 100% (51/51), done.
remote: Compressing objects: 100% (38/38), done.
remote: Total 51 (delta 17), reused 45 (delta 11), pack-reused 0
Receiving objects: 100% (51/51), 24.59 KiB | 6.15 MiB/s, done.
Resolving deltas: 100% (17/17), done.
```

Navigate to the Cloned Repository:

Definition: After cloning, use the cd command in your terminal or command prompt to navigate into the directory created by the cloning process. This directory contains all the project files and the entire Git repository history.

How to Navigate:

Use the cd command followed by the repository name to enter the repository directory. example:



```bash
cd repository
```

Result : Created and executed a Copy (or Clone) the Repository to the Local Machine that has been done successfully.

Exp. No : 6. Add a file to your local repo and "commit" (save) the changes.

Aim: To Add a File to Your Local Repository and Commit the Changes

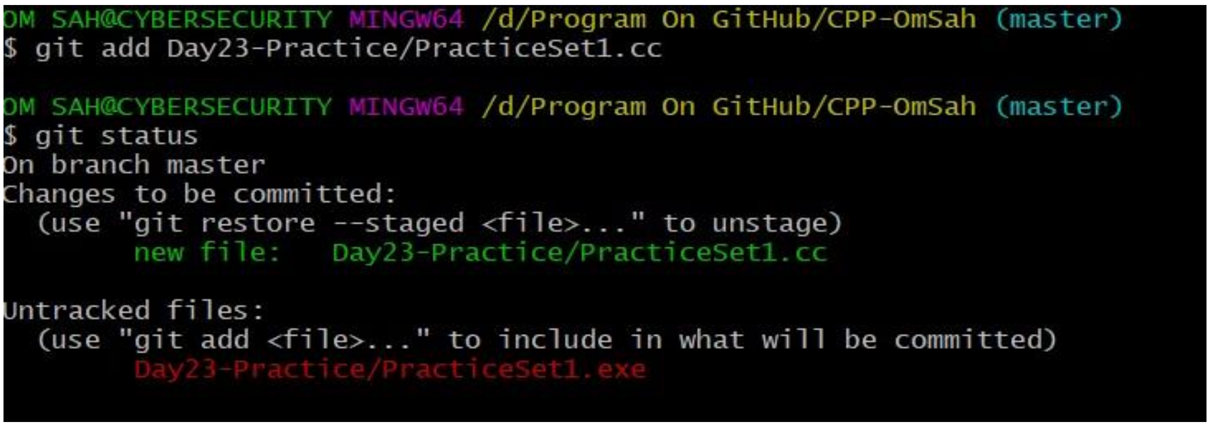
Definition: The aim of this step is to make changes to the files in your local Git repository and save those changes as a snapshot in Git's version history. This allows you to track and manage the project's development over time.

Add a File to Your Local Repository:

Definition: Before committing, you need to stage the changes you want to include in the next commit. Staging means marking specific files to be included in the commit. You can stage new files or changes to existing files.

How to Stage a File:

Use the git add command to stage one or more files. For example, to stage a single file:



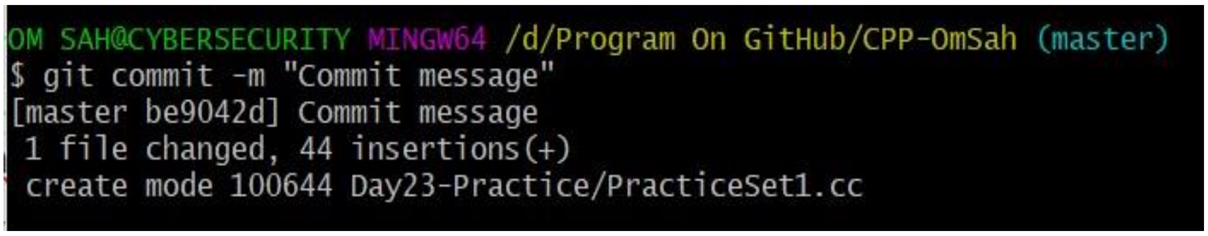
```
OM SAH@CYBERSECURITY MINGW64 /d/Program On GitHub/CPP-OmSah (master)
$ git add Day23-Practice/PracticeSet1.cc

OM SAH@CYBERSECURITY MINGW64 /d/Program On GitHub/CPP-OmSah (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   Day23-Practice/PracticeSet1.cc

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Day23-Practice/PracticeSet1.exe
```

Commit the Changes:

Definition: Committing records the staged changes as a new snapshot in the project's version history. Each commit has a unique identifier and a commit message that describes the changes made in that snapshot.

How to Commit:

Use the git commit command to create a commit. You should also include a meaningful commit message to describe message to changes you made in this commit.



```
OM SAH@CYBERSECURITY MINGW64 /d/Program On GitHub/CPP-OmSah (master)
$ git commit -m "Commit message"
[master be9042d] Commit message
 1 file changed, 44 insertions(+)
 create mode 100644 Day23-Practice/PracticeSet1.cc
```

Result : Added a File to the Local Repository and Commited the Changes made successfully.

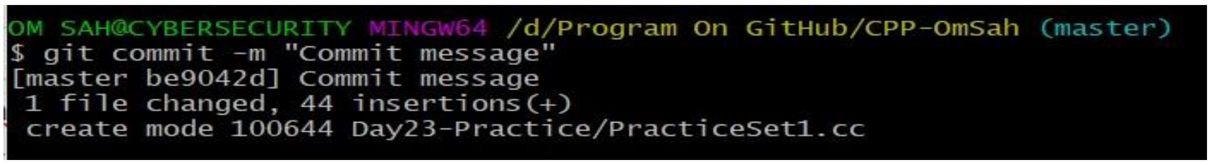
Exp. No : 7. "Push" your changes to your main branch

Aim: Synchronize Changes Made Locally with Remote Repository's Main Branch

Definition: Pushing changes refers to the action of sending committed changes from your local repository to the remote repository, specifically to a specified branch, often the main or master branch. This process ensures that the changes you made locally become visible to others collaborating on the project. Pushing is a fundamental operation in Git-based version control systems and is essential for collaborative development.

How to Push Changes:

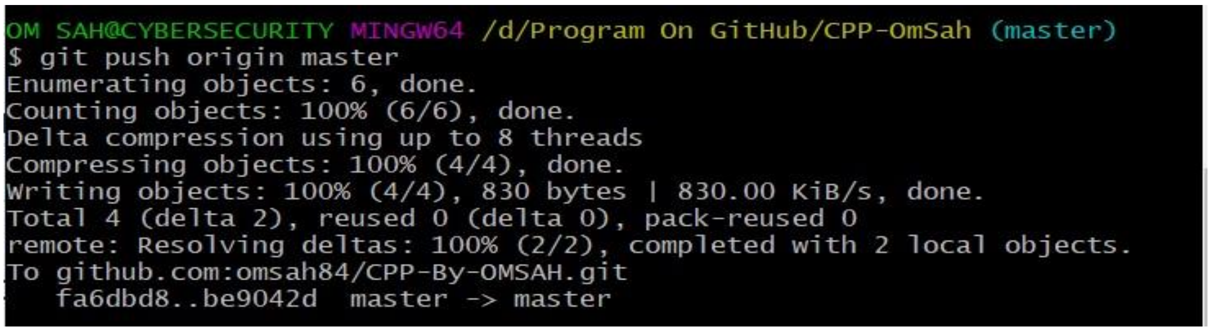
After making changes to files in your local repository, you need to commit those changes to the local repository. Use the following Git command to commit changes.



```
OM SAH@CYBERSECURITY MINGW64 /d/Program On GitHub/CPP-OmSah (master)
$ git commit -m "Commit message"
[master be9042d] Commit message
 1 file changed, 44 insertions(+)
 create mode 100644 Day23-Practice/PracticeSet1.cc
```

Replace "Your commit message here" with a descriptive message explaining the changes made in the commit.

Once changes are committed, you can push them to the main branch of the remote repository using the following Git command.



```
OM SAH@CYBERSECURITY MINGW64 /d/Program On GitHub/CPP-OmSah (master)
$ git push origin master
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 830 bytes | 830.00 KiB/s, done.
Total 4 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:omsah84/CPP-By-OMSAH.git
   fa6dbd8..be9042d  master -> master
```

In this command, origin represents the default name of the remote repository, and main is the name of the branch you want to push the changes to. If your main branch has a different name (such as master), replace main with the appropriate branch name.

Result : Synchronize Changes Made Locally with Remote Repository's Main Branch successfully.

Exp. No : 8. Make a change to your file with a git hosting tool and Commit.

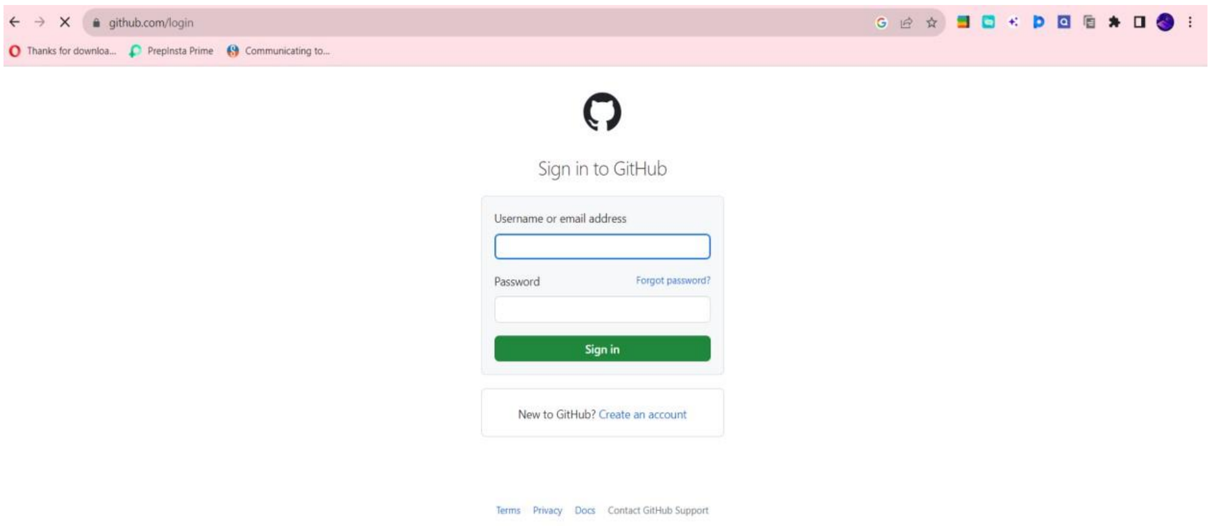
Aim: To modify a file stored in a Git repository hosted on a remote platform, such as GitHub, and then create a commit to record these changes in the repository's history.

Definition: Making changes and committing them in a Git hosting tool involves the process of updating a file's content, adding new files, or deleting files directly within the web interface of the Git hosting platform. After making the changes, you create a commit to document and track these modifications in the project's version history.

Procedure:

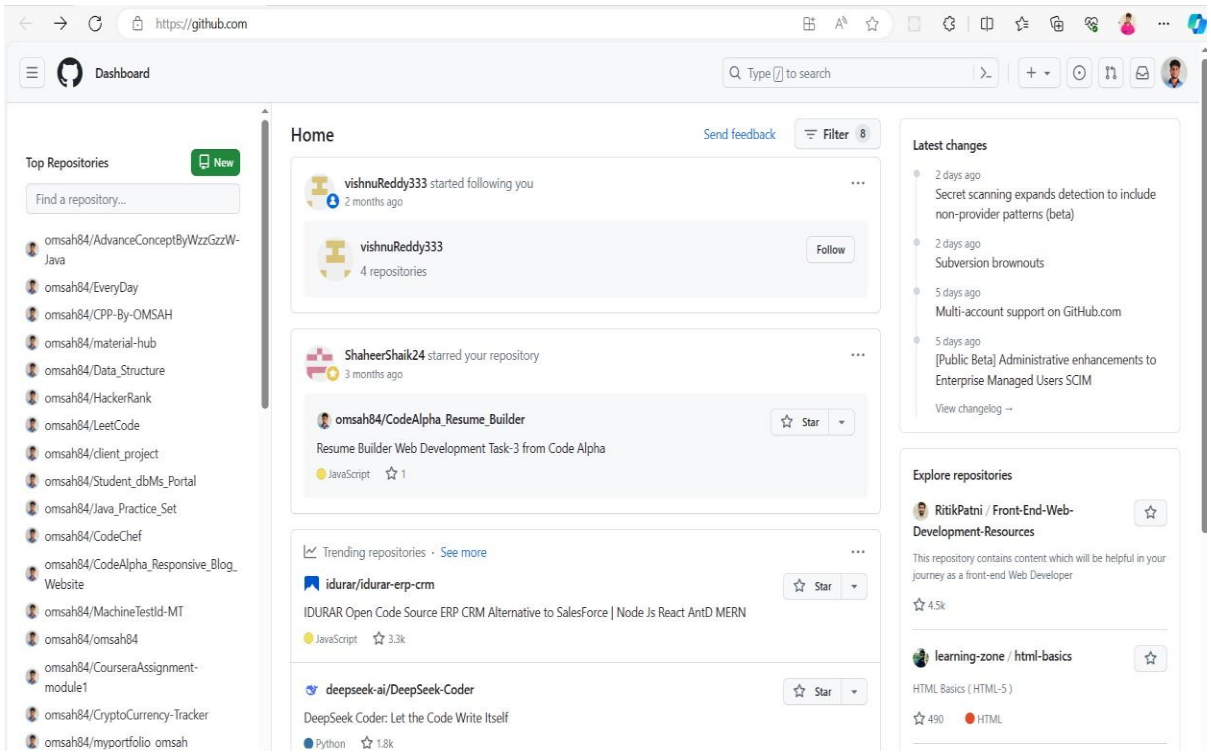
Log In to Your Git Hosting Tool:

Start by logging in to the Git hosting platform where your repository is hosted. For example, if your repository is on GitHub, visit GitHub and sign in to your account.



Navigate to the Repository:

Locate and select the repository in which you want to makechanges. .



https://github.com

Dashboard

Type / to search

Top Repositories New

Find a repository...

omsah84/AdvanceConceptByWzzGzzW-Java
omsah84/EveryDay
omsah84/CPP-By-OMSAH
omsah84/material-hub
omsah84/Data_Structure
omsah84/HackerRank
omsah84/LeetCode
omsah84/client_project
omsah84/Student_dbMs_Portal
omsah84/Java_Practice_Set
omsah84/CodeChef
omsah84/CodeAlpha_Responsive_Blog_Website
omsah84/MachineTestId-MT
omsah84/omsah84
omsah84/CourseraAssignment-module1
omsah84/CryptoCurrency-Tracker
omsah84/myportfolio_omsah

Home Send feedback Filter 8

vishnuReddy333 started following you
2 months ago
vishnuReddy333
4 repositories
Follow

ShaheerShaik24 starred your repository
3 months ago
omsah84/CodeAlpha_Resume_Builder
Resume Builder Web Development Task-3 from Code Alpha
JavaScript ☆ 1
Star

Trending repositories · See more
idurar/idurar-erp-crm
IDURAR Open Code Source ERP CRM Alternative to SalesForce | Node Js React AntD MERN
JavaScript ☆ 3.3k
Star

deepseek-ai/DeepSeek-Coder
DeepSeek Coder: Let the Code Write Itself
Python ☆ 1.8k
Star

Latest changes

2 days ago
Secret scanning expands detection to include non-provider patterns (beta)

2 days ago
Subversion brownouts

5 days ago
Multi-account support on GitHub.com

5 days ago
[Public Beta] Administrative enhancements to Enterprise Managed Users SCIM

View changelog →

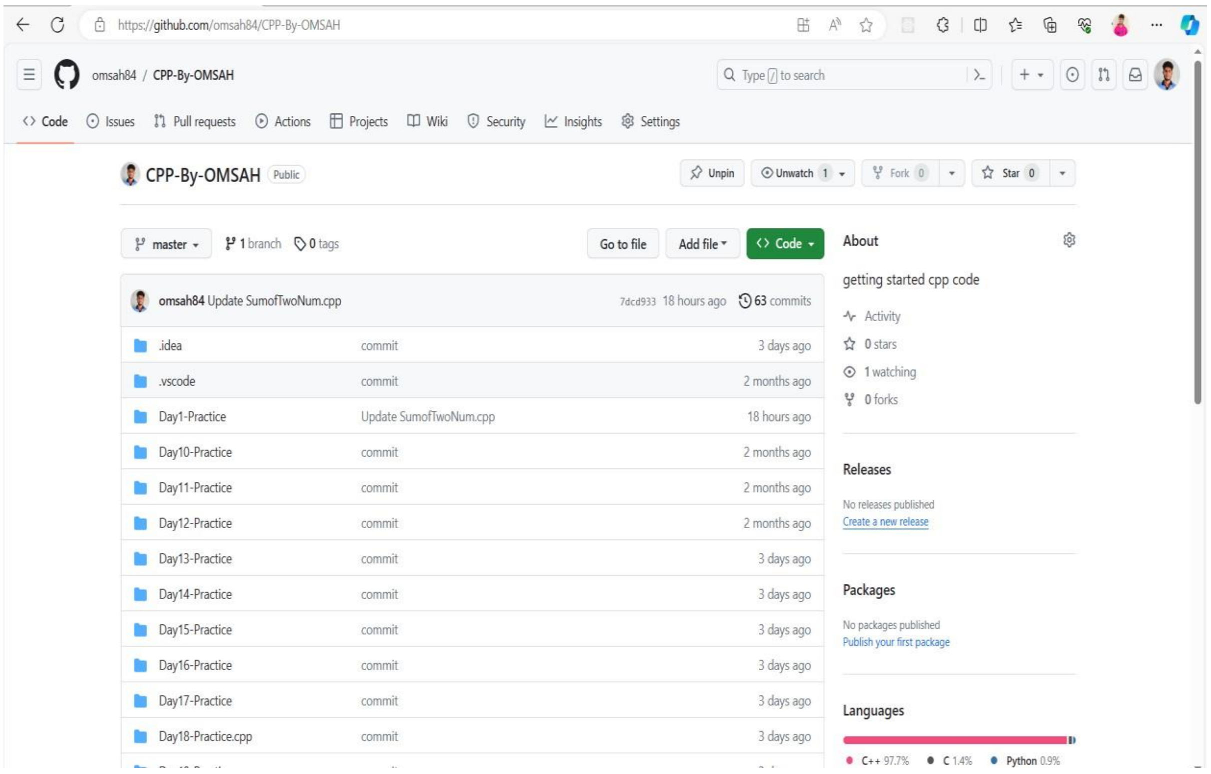
Explore repositories

RitikPatni / Front-End-Web-Development-Resources
This repository contains content which will be helpful in your journey as a front-end Web Developer
☆ 4.5k

learning-zone / html-basics
HTML Basics (HTML-5)
☆ 490 HTML

Find the File to Modify:

Use the platform's file navigation to locate the specific file you wish to modify.



https://github.com/omsah84/CPP-By-OMSAH

omsah84 / CPP-By-OMSAH

Type / to search

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

CPP-By-OMSAH Public

Unpin Unwatch 1 Fork 0 Star 0

master 1 branch 0 tags Go to file Add file Code

omsah84 Update SumofTwoNum.cpp 7dcd933 18 hours ago 63 commits

.idea	commit	3 days ago
.vscode	commit	2 months ago
Day1-Practice	Update SumofTwoNum.cpp	18 hours ago
Day10-Practice	commit	2 months ago
Day11-Practice	commit	2 months ago
Day12-Practice	commit	2 months ago
Day13-Practice	commit	3 days ago
Day14-Practice	commit	3 days ago
Day15-Practice	commit	3 days ago
Day16-Practice	commit	3 days ago
Day17-Practice	commit	3 days ago
Day18-Practice.cpp	commit	3 days ago

About
getting started cpp code
Activity
0 stars
1 watching
0 forks

Releases
No releases published
Create a new release

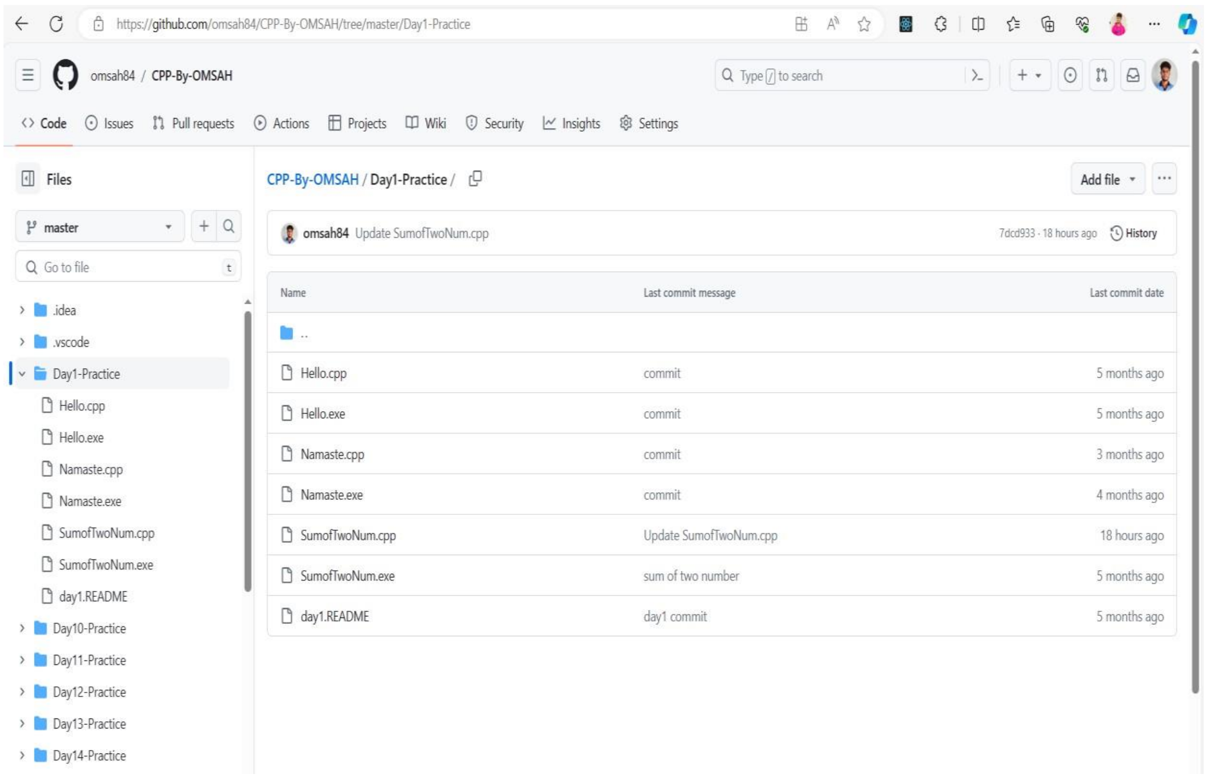
Packages
No packages published
Publish your first package

Languages
● C++ 97.7% ● C 1.4% ● Python 0.9%

Edit the File:

Click on the file's name to open it in an editor provided by the hosting platform.

Make the necessary changes to the file's content. You can use the platform's built-in text editor to make edits.



https://github.com/omsah84/CPP-By-OMSAH/tree/master/Day1-Practice

omsah84 / CPP-By-OMSAH

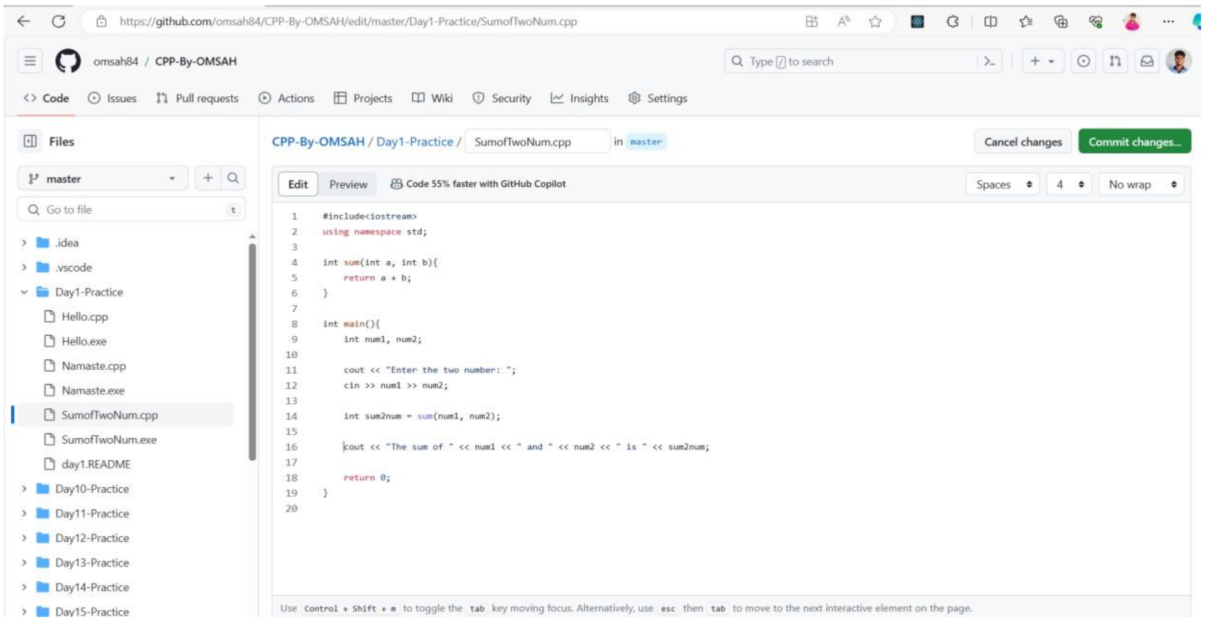
CPP-By-OMSAH / Day1-Practice /

omsah84 Update SumofTwoNum.cpp — 7dcd933 · 18 hours ago — History

Name	Last commit message	Last commit date
..		
Hello.cpp	commit	5 months ago
Hello.exe	commit	5 months ago
Namaste.cpp	commit	3 months ago
Namaste.exe	commit	4 months ago
SumofTwoNum.cpp	Update SumofTwoNum.cpp	18 hours ago
SumofTwoNum.exe	sum of two number	5 months ago
day1.README	day1 commit	5 months ago

Add a Commit Message:

After making your changes, you'll typically find an option to add a commit message. This message should describe the purpose of your changes. It's essential to provide a clear and informative message.



https://github.com/omsah84/CPP-By-OMSAH/edit/master/Day1-Practice/SumofTwoNum.cpp

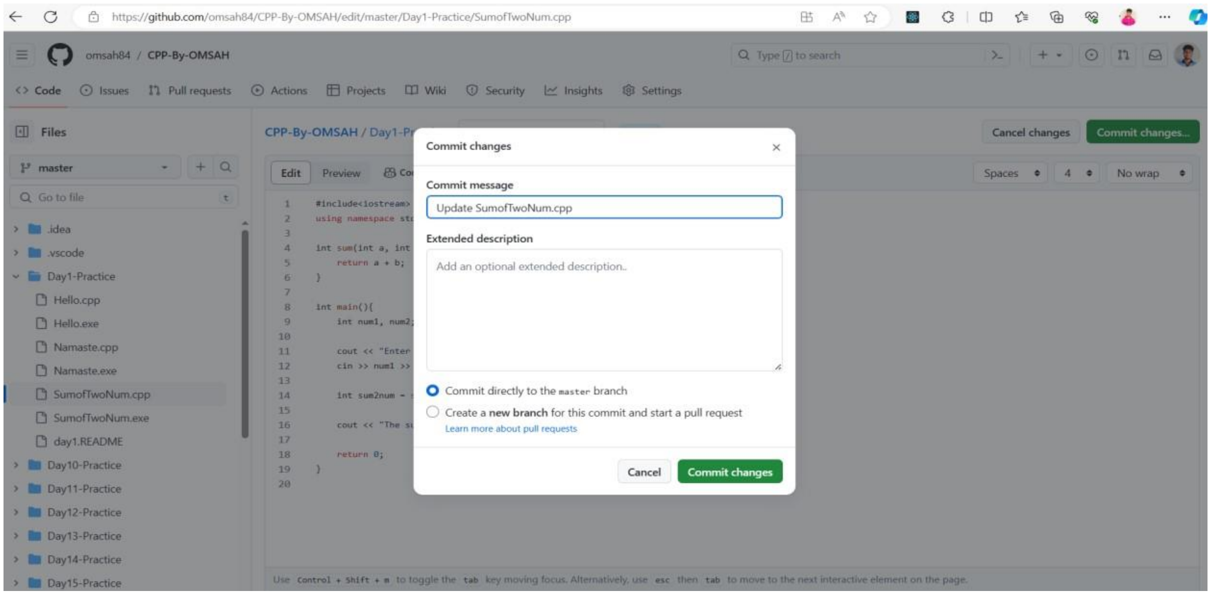
CPP-By-OMSAH / Day1-Practice / SumofTwoNum.cpp in master

```cpp
#include<iostream>
using namespace std;

int sum(int a, int b){
    return a + b;
}

int main(){
    int num1, num2;

    cout << "Enter the two number: ";
    cin >> num1 >> num2;

    int sum2num = sum(num1, num2);

    cout << "The sum of " << num1 << " and " << num2 << " is " << sum2num;

    return 0;
}

```

Use Control + Shift + m to toggle the tab key moving focus. Alternatively, use esc then tab to move to the next interactive element on the page.

Commit the Changes:
Look for a “Commit” or “Save” button within the web interface.

Click this button to commit the changes to the remote repository.


https://github.com/omsah84/CPP-By-OMSAH/edit/master/Day1-Practice/SumofTwoNum.cpp

```
#include<iostream>
using namespace st

int sum(int a, int
    return a + b;
}

int main(){
    int num1, num2;

    cout << "Enter
    cin >> num1 >>

    int sum2num =

    cout << "The s

    return 0;
}
```

Commit changes

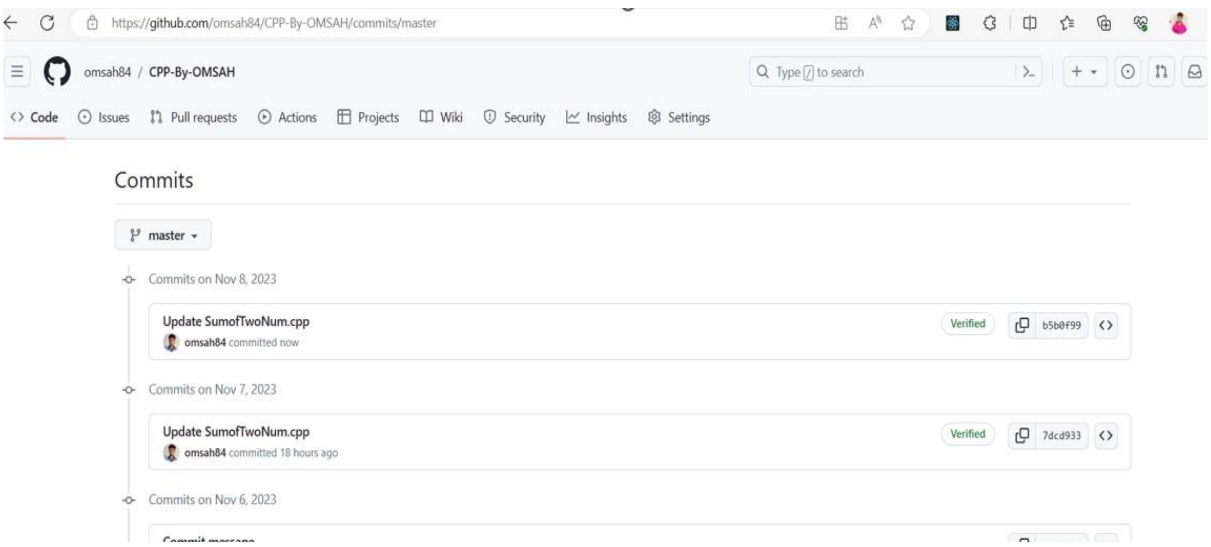
Commit message: Update SumofTwoNum.cpp

Extended description: Add an optional extended description..

[✓ Commit directly to the `master` branch]
Create a **new branch** for this commit and start a pull request
Learn more about pull requests

Use `Control + Shift + m` to toggle the `tab` key moving focus. Alternatively, use `esc` then `tab` to move to the next interactive element on the page.

Verify the Commit:
After committing, you can usually view the commit details, including the commit message and a unique commit hash, in the platform's interface.


https://github.com/omsah84/CPP-By-OMSAH/commits/master

Commits

master

Commits on Nov 8, 2023

Update SumofTwoNum.cpp
omsah84 committed now — Verified — b5b0f99

Commits on Nov 7, 2023

Update SumofTwoNum.cpp
omsah84 committed 18 hours ago — Verified — 7dcd933

Commits on Nov 6, 2023

Result : Changes in the file with a git hosting tool and commit has been executed successfully.

Exp. No : 9. "Pull" the changes to your local machine.

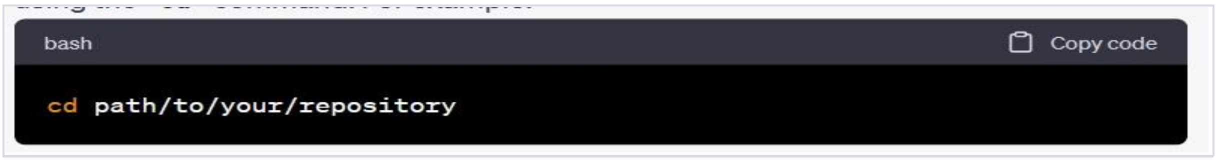
Aim: Update Your Local Repository with Changes from the Remote Repository.

Definition: Pulling changes in Git refers to the process of fetching and merging changes from a remote repository into your local repository. When other collaborators make changes to the remote repository (for example, by pushing their changes), pulling allows you to incorporate these updates into your own local working copy. This ensures that your local files are up-to-date with the latest changes made by others.

Procedure to Pull Changes:

Navigate to Your Local Repository:

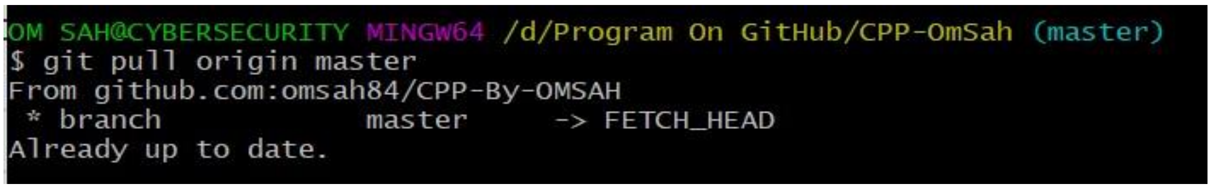
Open your terminal or command prompt and navigate to the local repository directory using the cd command. For example:



```bash
cd path/to/your/repository
```

Pull Changes from the Remote Repository:

Use the following Git command to pull changes from the remote repository into your local branch (e.g., main):

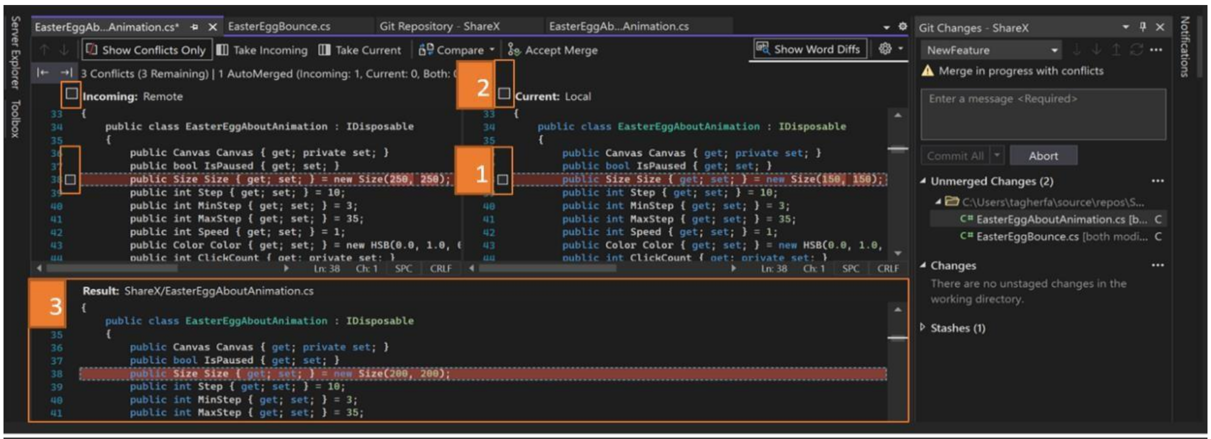


```
OM SAH@CYBERSECURITY MINGW64 /d/Program On GitHub/CPP-OmSah (master)
$ git pull origin master
From github.com:omsah84/CPP-By-OMSAH
 * branch            master     -> FETCH_HEAD
Already up to date.
```

In this command, origin represents the default name of the remote repository, and main is the branch from which you want to pull changes. If your main branch has a different name (such as master), replace main with the appropriate branch name.

Resolve any Merge Conflicts (if necessary):

If Git detects that there are conflicting changes between your local branch and the remote branch, you'll need to resolve these conflicts manually before you can complete the pull operation. Git will mark the conflicting files, and you'll need to edit them to resolve the differences.


EasterEggAb...Animation.cs*
EasterEggBounce.cs
Git Repository - ShareX
EasterEggAb...Animation.cs
Show Conflicts Only
Take Incoming
Take Current
Compare
Accept Merge
Show Word Diffs
3 Conflicts (3 Remaining) | 1 AutoMerged (Incoming: 1, Current: 0, Both:
2
1
3
Ln: 38 Ch: 1 SPC CRLF
Git Changes - ShareX
NewFeature
Merge in progress with conflicts
Enter a message <Required>
Commit All
Abort
Unmerged Changes (2)
C:\Users\tagherfa\source\repos\S...
EasterEggAboutAnimation.cs [b...
EasterEggBounce.cs [both modi...
Changes
There are no unstaged changes in the working directory.
Stashes (1)
Server Explorer
Toolbox
Notifications

Incoming: Remote

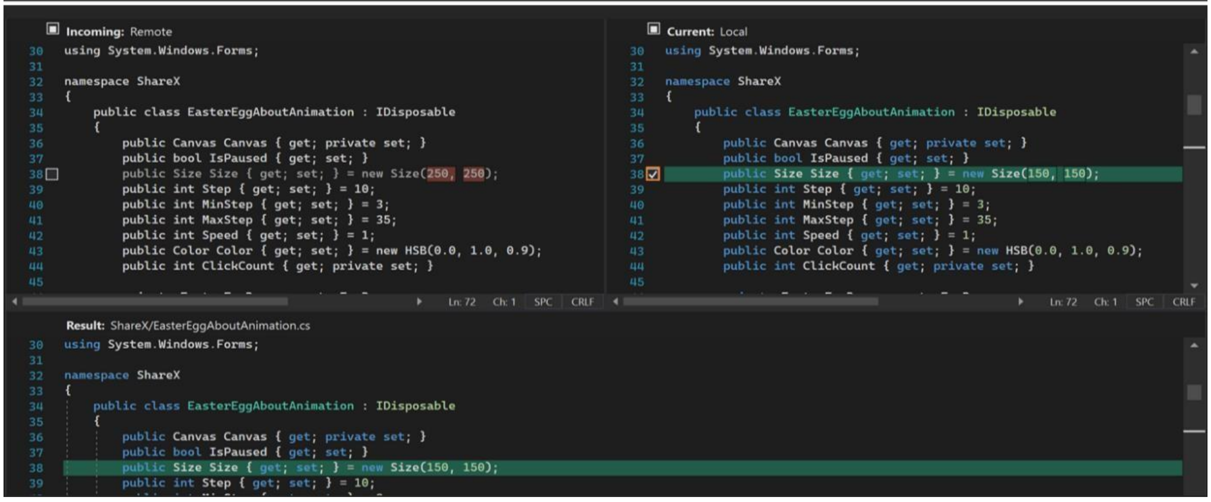
```
{
    public class EasterEggAboutAnimation : IDisposable
    {
        public Canvas Canvas { get; private set; }
        public bool IsPaused { get; set; }
        public Size Size { get; set; } = new Size(250, 250);
        public int Step { get; set; } = 10;
        public int MinStep { get; set; } = 3;
        public int MaxStep { get; set; } = 35;
        public int Speed { get; set; } = 1;
        public Color Color { get; set; } = new HSB(0.0, 1.0, (
        public int ClickCount { get; private set; }
```

Current: Local

```
{
    public class EasterEggAboutAnimation : IDisposable
    {
        public Canvas Canvas { get; private set; }
        public bool IsPaused { get; set; }
        public Size Size { get; set; } = new Size(150, 150);
        public int Step { get; set; } = 10;
        public int MinStep { get; set; } = 3;
        public int MaxStep { get; set; } = 35;
        public int Speed { get; set; } = 1;
        public Color Color { get; set; } = new HSB(0.0, 1.0,
        public int ClickCount { get; private set; }
```

Result: ShareX/EasterEggAboutAnimation.cs

```
{
    public class EasterEggAboutAnimation : IDisposable
    {
        public Canvas Canvas { get; private set; }
        public bool IsPaused { get; set; }
        public Size Size { get; set; } = new Size(200, 200);
        public int Step { get; set; } = 10;
        public int MinStep { get; set; } = 3;
        public int MaxStep { get; set; } = 35;
```



Incoming: Remote

```
using System.Windows.Forms;

namespace ShareX
{
    public class EasterEggAboutAnimation : IDisposable
    {
        public Canvas Canvas { get; private set; }
        public bool IsPaused { get; set; }
        public Size Size { get; set; } = new Size(250, 250);
        public int Step { get; set; } = 10;
        public int MinStep { get; set; } = 3;
        public int MaxStep { get; set; } = 35;
        public int Speed { get; set; } = 1;
        public Color Color { get; set; } = new HSB(0.0, 1.0, 0.9);
        public int ClickCount { get; private set; }
```

Current: Local

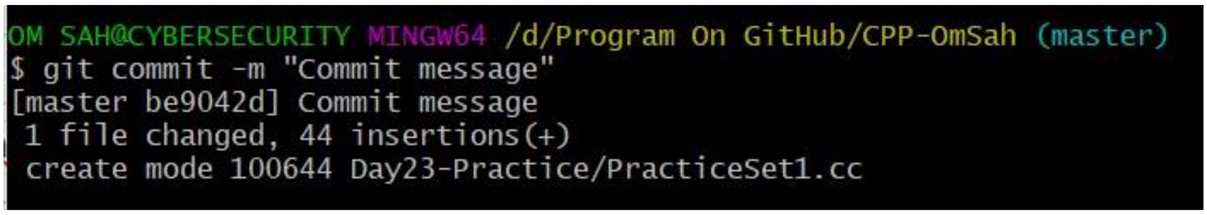
```
using System.Windows.Forms;

namespace ShareX
{
    public class EasterEggAboutAnimation : IDisposable
    {
        public Canvas Canvas { get; private set; }
        public bool IsPaused { get; set; }
        public Size Size { get; set; } = new Size(150, 150);
        public int Step { get; set; } = 10;
        public int MinStep { get; set; } = 3;
        public int MaxStep { get; set; } = 35;
        public int Speed { get; set; } = 1;
        public Color Color { get; set; } = new HSB(0.0, 1.0, 0.9);
        public int ClickCount { get; private set; }
```

Result: ShareX/EasterEggAboutAnimation.cs

```
using System.Windows.Forms;

namespace ShareX
{
    public class EasterEggAboutAnimation : IDisposable
    {
        public Canvas Canvas { get; private set; }
        public bool IsPaused { get; set; }
        public Size Size { get; set; } = new Size(150, 150);
        public int Step { get; set; } = 10;
```

Commit the Merged Changes (if conflicts were resolved):

After resolving merge conflicts, you need to commit the merged changes locally before you can continue working. Use the git commit command to commit the resolved changes.



```
OM SAH@CYBERSECURITY MINGW64 /d/Program On GitHub/CPP-OmSah (master)
$ git commit -m "Commit message"
[master be9042d] Commit message
 1 file changed, 44 insertions(+)
 create mode 100644 Day23-Practice/PracticeSet1.cc
```

Result : Updating the Local Repository with Changes from the Remote Repository has been done successfully.

Exp. No : 10. Create a "branch" (version), make a change, commit the change.

Aim: Create an isolated development environment for a specific task, make changes to project files, and commit those changes to track the task's progress.

Definition: Create a Branch: In Git, a branch is a separate line of development. Creating a branch allows you to work on a specific task or feature without affecting the main project until the task is complete and ready to be integrated. Branching helps in parallel development and collaboration.

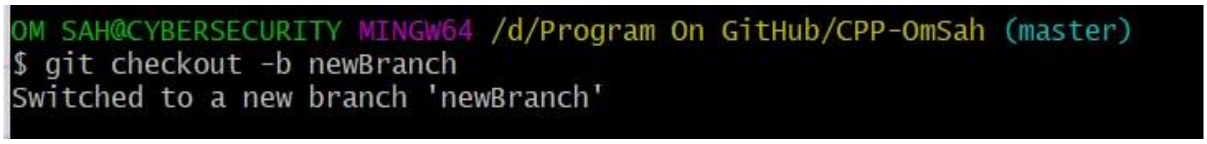
Make a Change: This involves modifying or adding code, files, or content within the project. The aim is to implement a specific task or feature as part of the development process.

Commit the Change: Committing changes in Git is the act of saving your work and creating a record of the modifications made in a branch. Each commit represents a point in the project's history and includes a commit message describing the changes made.

Procedure:

Create a Branch:

Use the following Git command to create a new branch. Replace <branch_name> with a descriptive name for your branch:



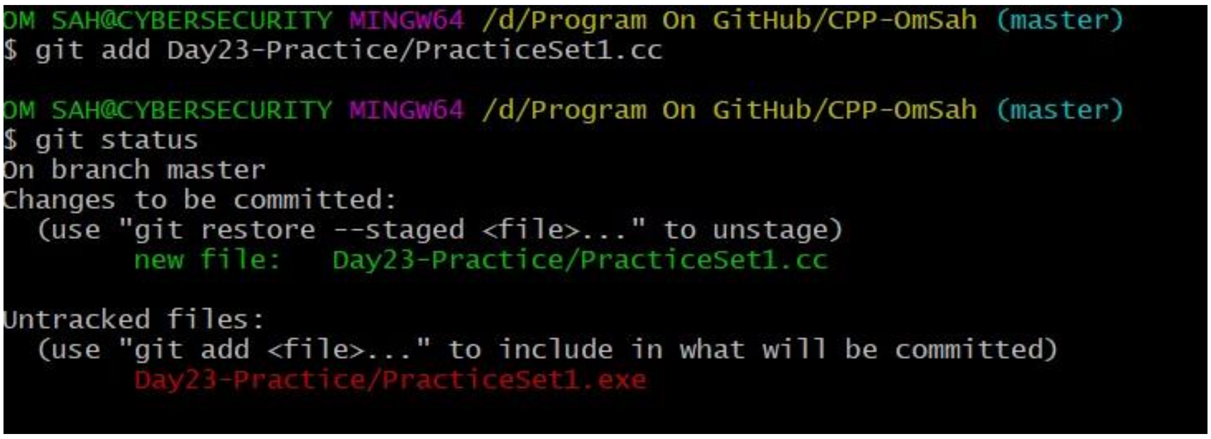
```
OM SAH@CYBERSECURITY MINGW64 /d/Program On GitHub/CPP-OmSah (master)
$ git checkout -b newBranch
Switched to a new branch 'newBranch'
```

Make a Change:

Modify or add files as necessary to implement the specific task or feature you're working on. You can use text editors or integrated development environments (IDEs) to make these changes.

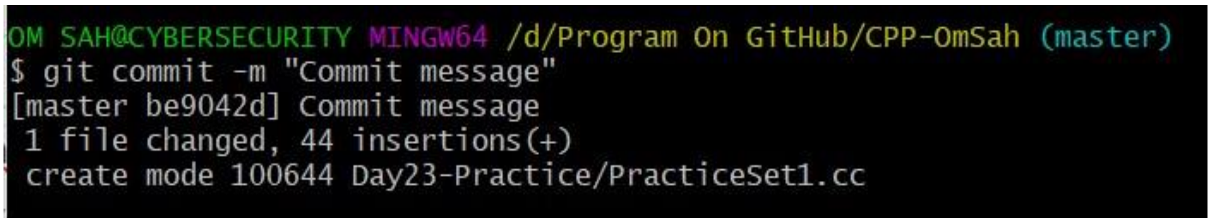
Stage Changes:

Use the git add command to stage the changes you've made. This prepares the changes for the next commit.



```
OM SAH@CYBERSECURITY MINGW64 /d/Program On GitHub/CPP-OmSah (master)
$ git add Day23-Practice/PracticeSet1.cc

OM SAH@CYBERSECURITY MINGW64 /d/Program On GitHub/CPP-OmSah (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   Day23-Practice/PracticeSet1.cc

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Day23-Practice/PracticeSet1.exe
```

Commit the Change:
Commit the staged changes with a descriptive commit message. This records the changes made to the branch.



```
OM SAH@CYBERSECURITY MINGW64 /d/Program On GitHub/CPP-OmSah (master)
$ git commit -m "Commit message"
[master be9042d] Commit message
 1 file changed, 44 insertions(+)
 create mode 100644 Day23-Practice/PracticeSet1.cc
```

Result : Created a "branch", make a change, commit the change has been successfully completed.

Exp. No : 11. Open a "pull request" (propose changes to the main branch).

Aim: To create Propose and Discuss Changes Made in a Feature or Topic Branch with the Repository's Maintainers and Collaborators.

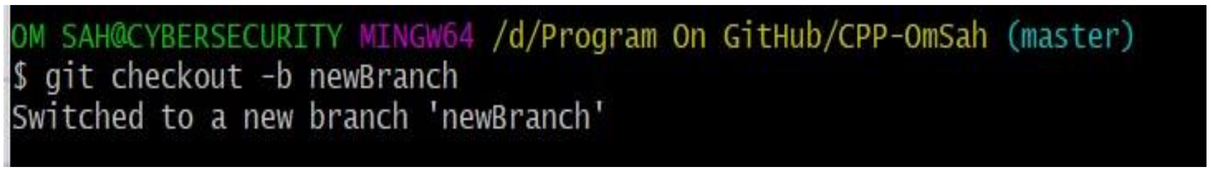
Definition: A pull request (PR) is a way to request that changes you've made in a feature or topic branch be reviewed, discussed, and potentially merged into the main branch of a Git repository. It serves as a formal mechanism for collaborating on code changes and allows maintainers and collaborators to provide feedback and ensure code quality before integrating the changes.

Procedure:

Create a Feature Branch:

Before making changes, create a new branch from the main branch of the repository. This feature branch will contain the changes you want to propose in the pull request.

Use the following Git command to create a new branch:

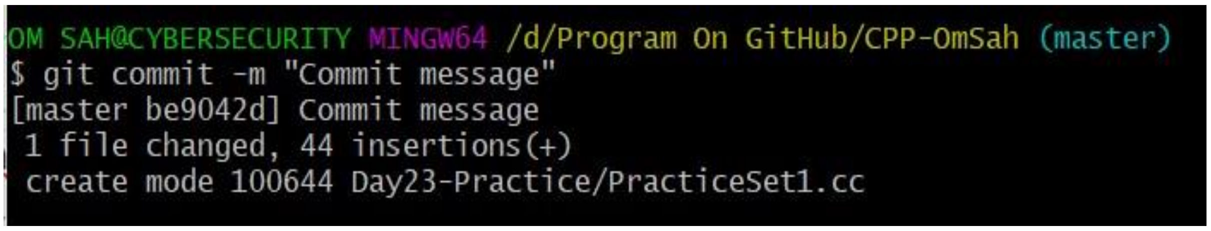


```
OM SAH@CYBERSECURITY MINGW64 /d/Program On GitHub/CPP-OmSah (master)
$ git checkout -b newBranch
Switched to a new branch 'newBranch'
```

Make and Commit Changes:

Make the necessary code changes in your feature branch.

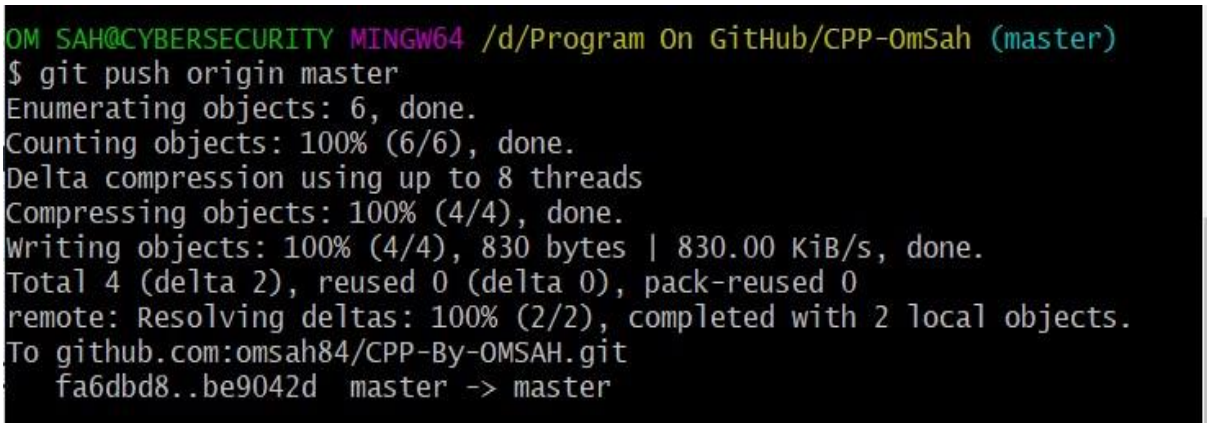
Commit your changes with descriptive commit messages using the git commit command:



```
OM SAH@CYBERSECURITY MINGW64 /d/Program On GitHub/CPP-OmSah (master)
$ git commit -m "Commit message"
[master be9042d] Commit message
 1 file changed, 44 insertions(+)
 create mode 100644 Day23-Practice/PracticeSet1.cc
```

Push the Feature Branch:

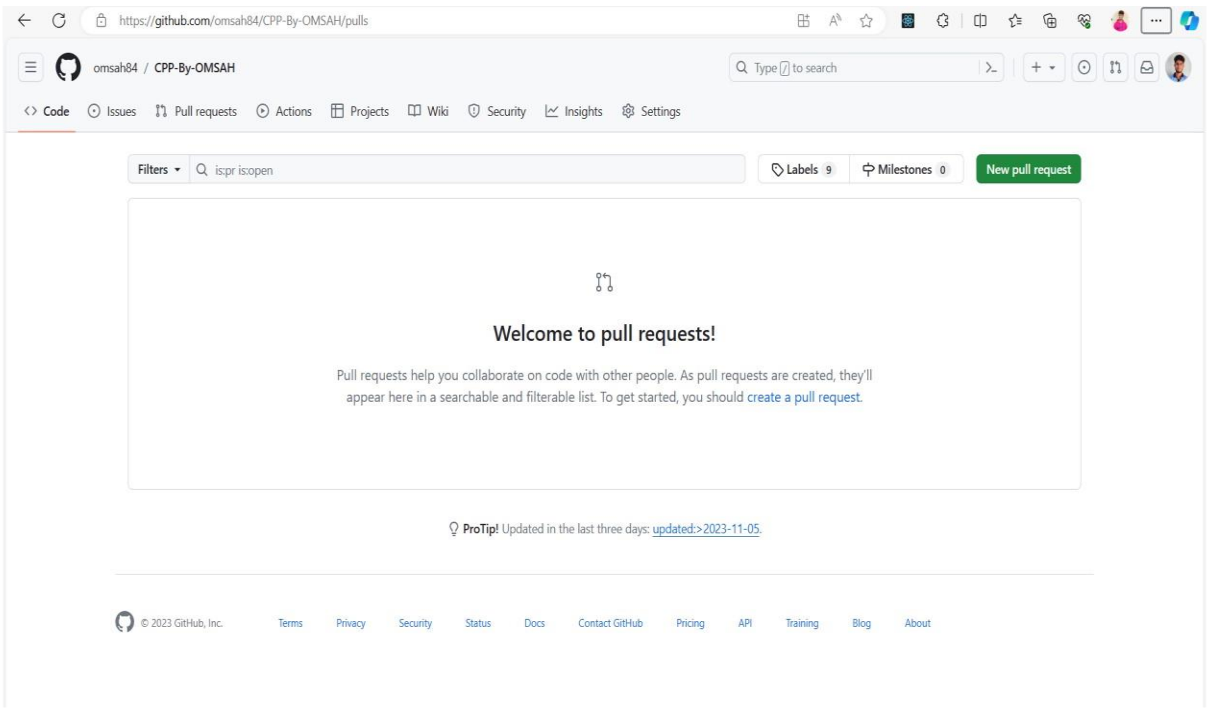
Push your feature branch with the changes to the remote repository using:



```
OM SAH@CYBERSECURITY MINGW64 /d/Program On GitHub/CPP-OmSah (master)
$ git push origin master
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 830 bytes | 830.00 KiB/s, done.
Total 4 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:omsah84/CPP-By-OMSAH.git
   fa6dbd8..be9042d  master -> master
```

Open a Pull Request:
Visit the repository's page on the Git hosting platform (e.g., GitHub) where you cloned the repository.

There should be an option to "Create a Pull Request" or "New Pull Request." Click on it.



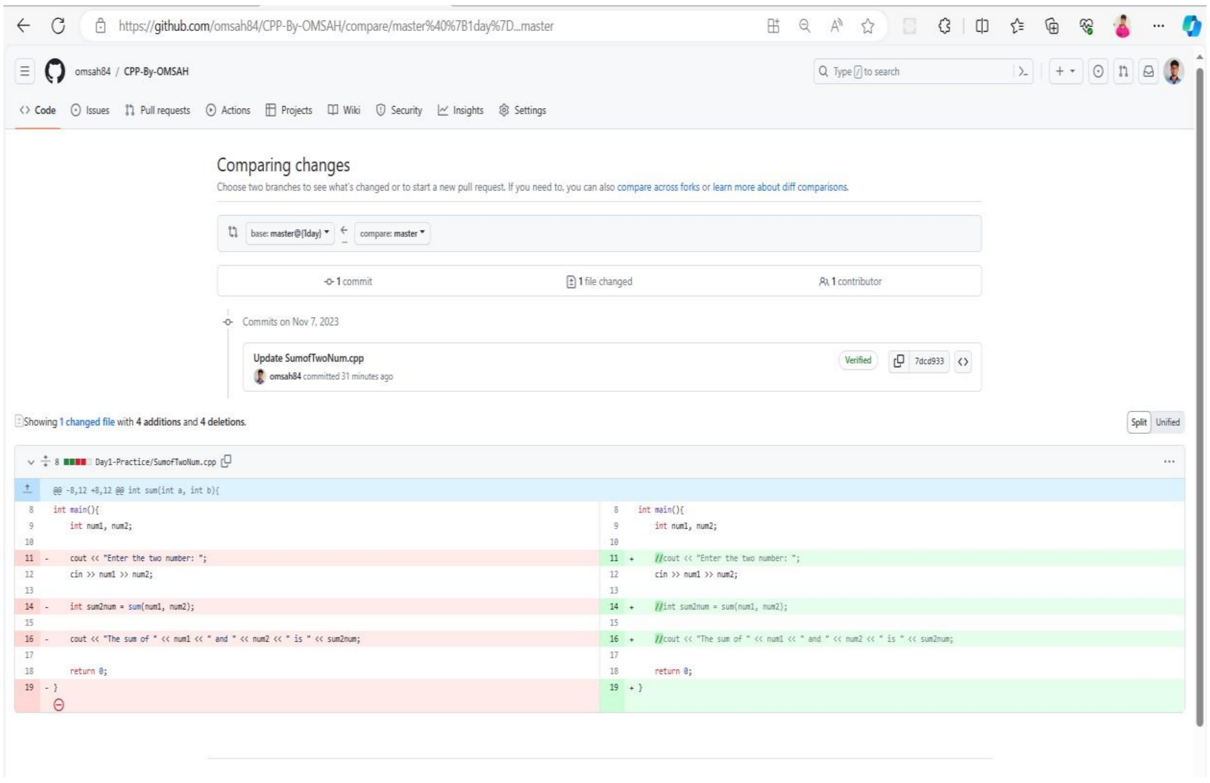
Welcome to pull requests!

Pull requests help you collaborate on code with other people. As pull requests are created, they'll appear here in a searchable and filterable list. To get started, you should create a pull request.

ProTip! Updated in the last three days: updated:>2023-11-05.

Compare and Review:
In the pull request interface, select the base branch (usually the main branch) and the compare branch (your feature branch).

Describe the changes you've made and the purpose of the pull request.



https://github.com/omsah84/CPP-By-OMSAH/compare/master%40%7B1day%7D...master

omsah84 / CPP-By-OMSAH

Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also compare across forks or learn more about diff comparisons.

base: master@{1day} ← compare: master

1 commit | 1 file changed | 1 contributor

Commits on Nov 7, 2023

Update SumofTwoNum.cpp
omsah84 committed 31 minutes ago — Verified — 7dcd933

Showing 1 changed file with 4 additions and 4 deletions.

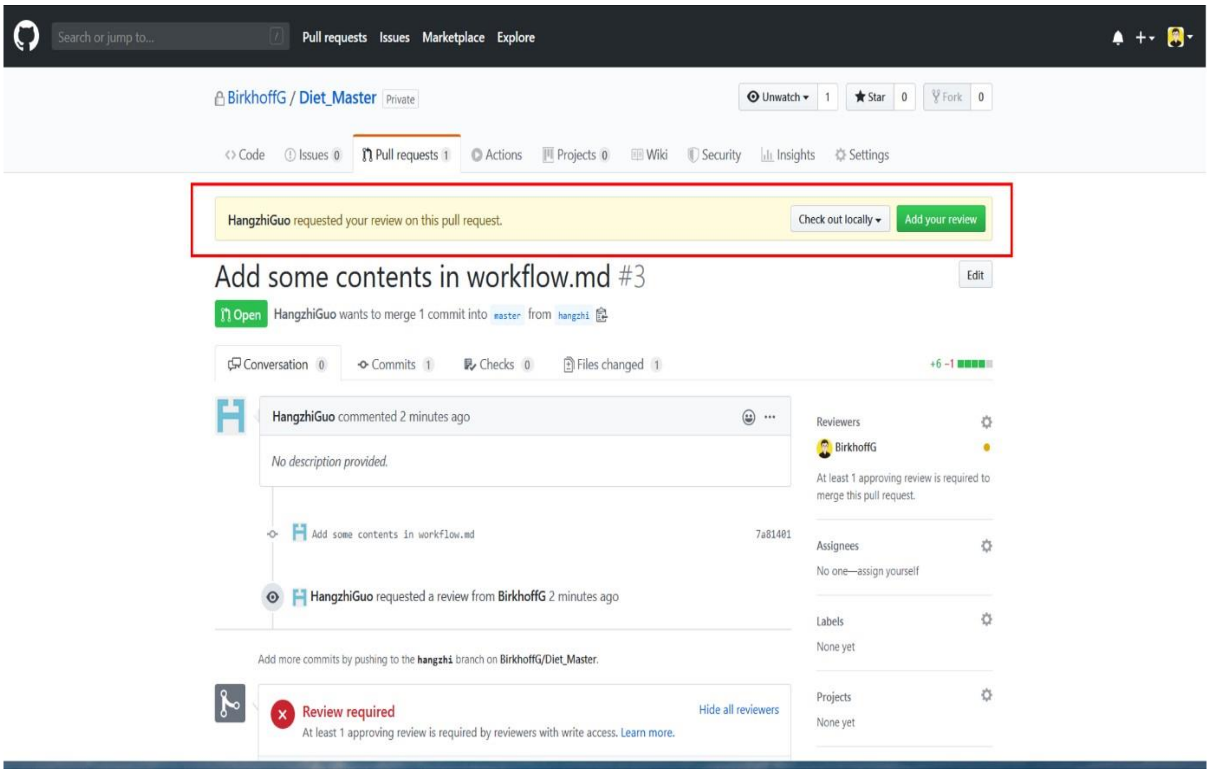
8 Day1-Practice/SumofTwoNum.cpp

```
@@ -8,12 +8,12 @@ int sum(int a, int b){
 int main(){
     int num1, num2;

-    cout << "Enter the two number: ";
     cin >> num1 >> num2;

-    int sum2num = sum(num1, num2);

-    cout << "The sum of " << num1 << " and " << num2 << " is " << sum2num;

     return 0;
- }
```

```
 int main(){
     int num1, num2;

+    //cout << "Enter the two number: ";
     cin >> num1 >> num2;

+    //int sum2num = sum(num1, num2);

+    //cout << "The sum of " << num1 << " and " << num2 << " is " << sum2num;

     return 0;
+ }
```

Request Reviewers:

If you want specific individuals to review your code, you can request their review on the pull request.



BirkhoffG / Diet_Master Private

HangzhiGuo requested your review on this pull request.

Add some contents in workflow.md #3

Open — HangzhiGuo wants to merge 1 commit into master from hangzhi

HangzhiGuo commented 2 minutes ago

No description provided.

Add some contents in workflow.md — 7a81401

HangzhiGuo requested a review from BirkhoffG 2 minutes ago

Add more commits by pushing to the hangzhi branch on BirkhoffG/Diet_Master.

Review required
At least 1 approving review is required by reviewers with write access. Learn more.

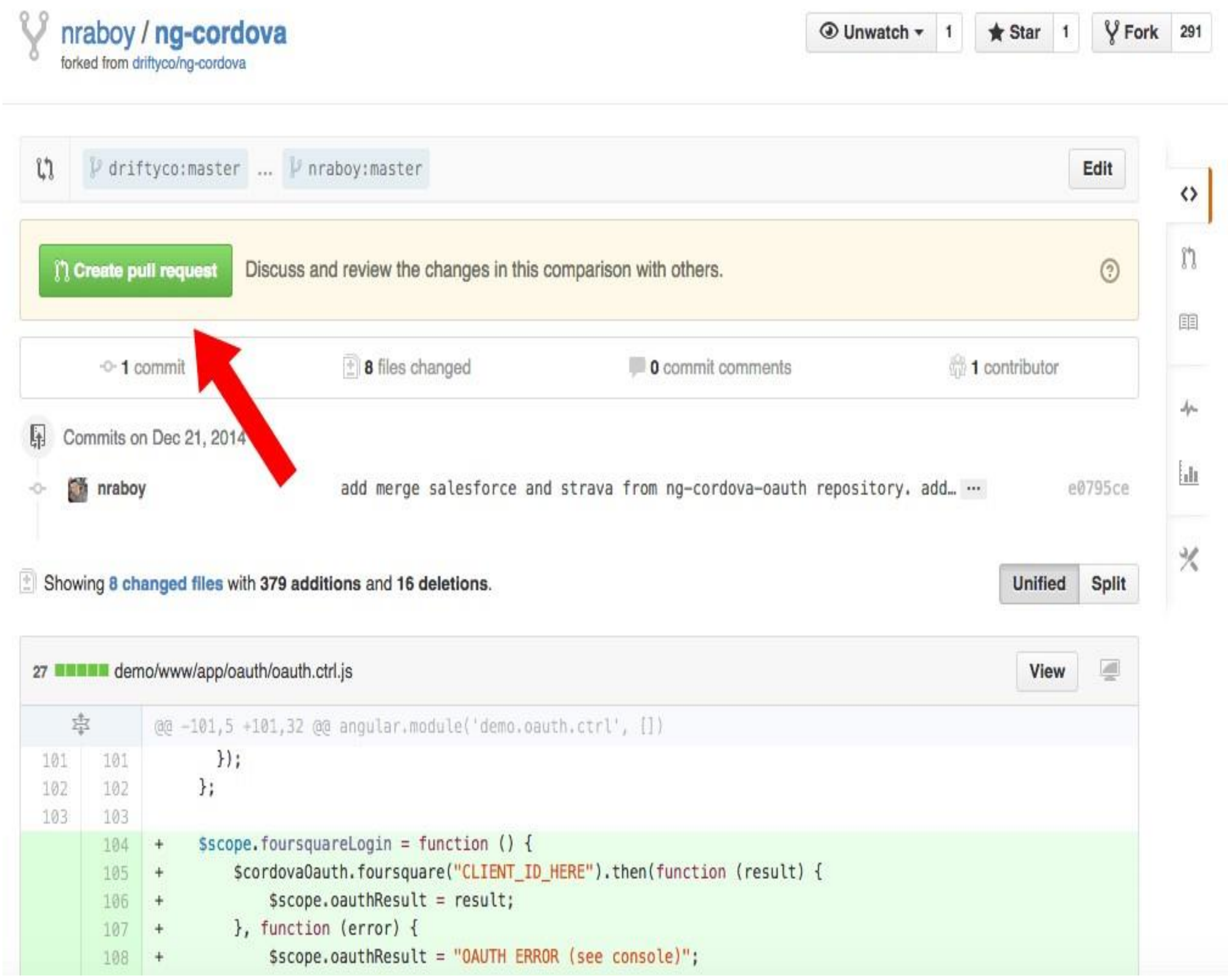
Reviewers: BirkhoffG
At least 1 approving review is required to merge this pull request.

Assignees: No one—assign yourself

Labels: None yet

Projects: None yet

Submit the Pull Request:
Finally, submit the pull request. This will notify the repository's maintainers and collaborators of the proposed changes.



Discuss and review the changes in this comparison with others.

1 commit 8 files changed 0 commit comments 1 contributor

Commits on Dec 21, 2014

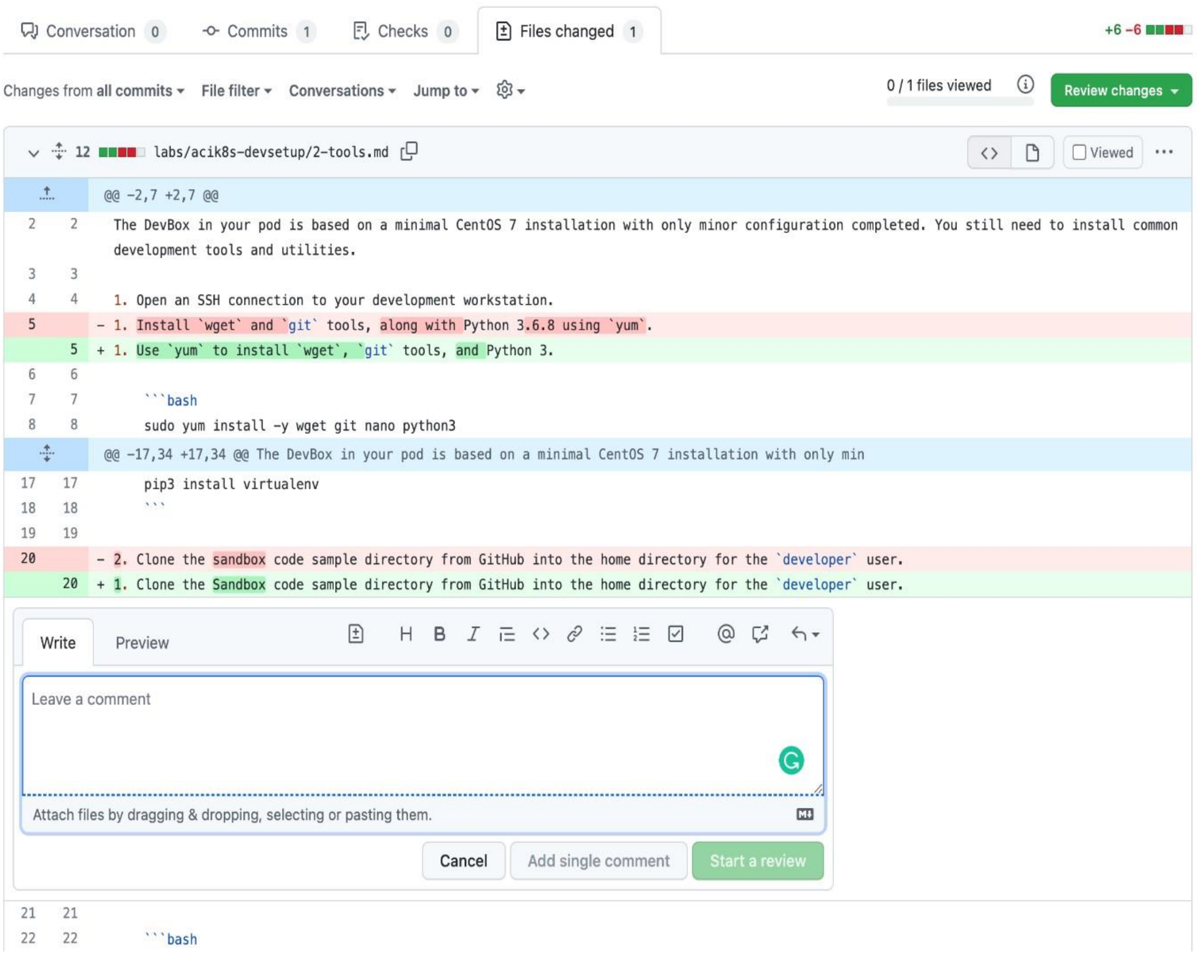
nraboy add merge salesforce and strava from ng-cordova-oauth repository. add… e0795ce

Showing 8 changed files with 379 additions and 16 deletions.

27 demo/www/app/oauth/oauth.ctrl.js

```
@@ -101,5 +101,32 @@ angular.module('demo.oauth.ctrl', [])
         });
     };

+    $scope.foursquareLogin = function () {
+        $cordovaOauth.foursquare("CLIENT_ID_HERE").then(function (result) {
+            $scope.oauthResult = result;
+        }, function (error) {
+            $scope.oauthResult = "OAUTH ERROR (see console)";
```

Discuss and Review:
Reviewers will examine the code changes, leave comments, and have discussions with you about the proposed modifications.



~~~
@@ -2,7 +2,7 @@
 The DevBox in your pod is based on a minimal CentOS 7 installation with only minor configuration completed. You still need to install common development tools and utilities.

 1. Open an SSH connection to your development workstation.
- 1. Install `wget` and `git` tools, along with Python 3.6.8 using `yum`.
+ 1. Use `yum` to install `wget`, `git` tools, and Python 3.

     ```bash
     sudo yum install -y wget git nano python3
@@ -17,34 +17,34 @@ The DevBox in your pod is based on a minimal CentOS 7 installation with only min
     pip3 install virtualenv
     ~~~

- 2. Clone the sandbox code sample directory from GitHub into the home directory for the `developer` user.
+ 1. Clone the Sandbox code sample directory from GitHub into the home directory for the `developer` user.

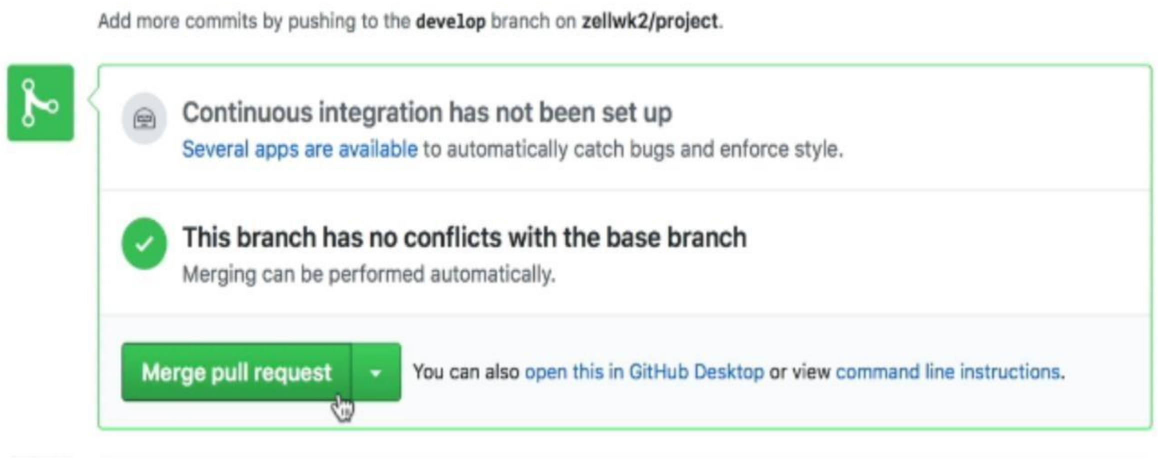
     ```bash
```

Make Necessary Changes:

Address any feedback or comments provided during the review process by making additional commits in the feature branch.

Merge the Pull Request:

Once the reviewers are satisfied with the changes and have approved the pull request, it can be merged into the main branch.



Add more commits by pushing to the **develop** branch on **zellwk2/project**.

Continuous integration has not been set up

Several apps are available to automatically catch bugs and enforce style.

This branch has no conflicts with the base branch

Merging can be performed automatically.

You can also open this in GitHub Desktop or view command line instructions.

Result : Pull request command has been executed successfully.

Exp. No : 12. "Merge" your branch to the main branch.

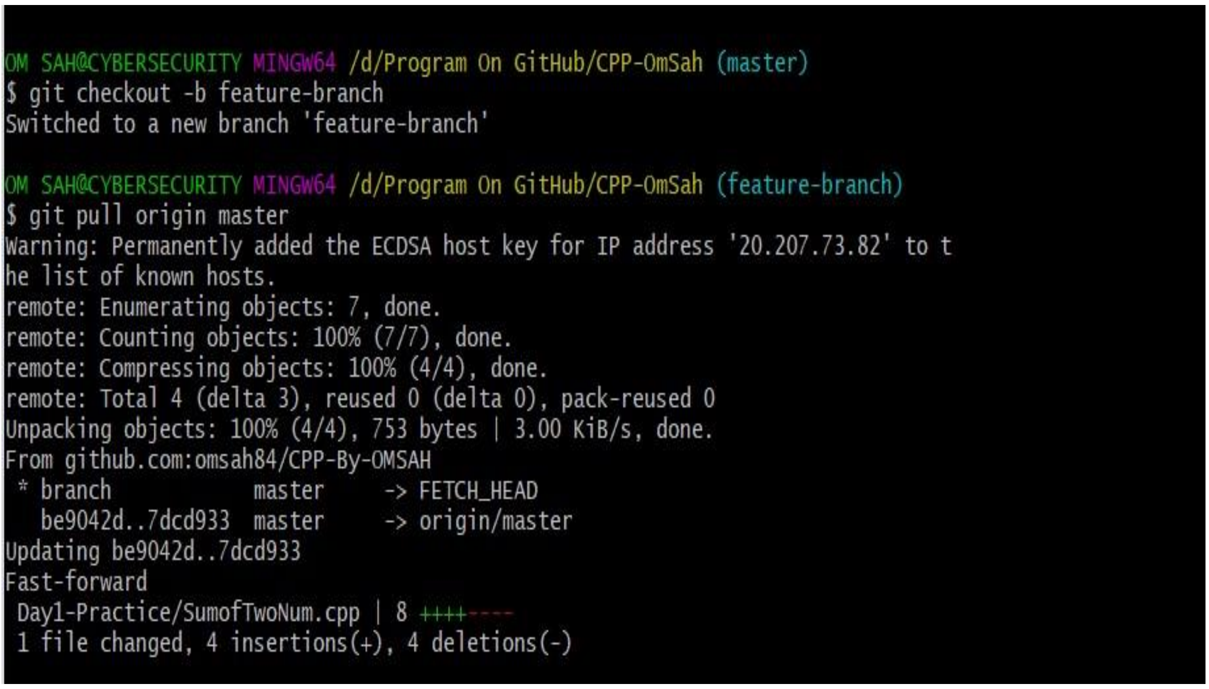
Aim: Integrate Changes Made in a Feature or Topic Branch into the Main Codebase.

Definition: Merging is the process of combining the changes made in a feature or topic branch back into the main branch (or any target branch). It integrates the new code, commits, and file changes from the source branch into the destination branch. Merging is a fundamental operation in Git, allowing teams to collaborate on different branches and consolidate their work into a unified codebase.

Procedure:

Ensure Your Branch is Up to Date:

Before merging, it's a good practice to ensure your feature branch is up to date with the latest changes from the main branch. You can do this by switching to your feature branch and pulling the latest changes from the master branch:

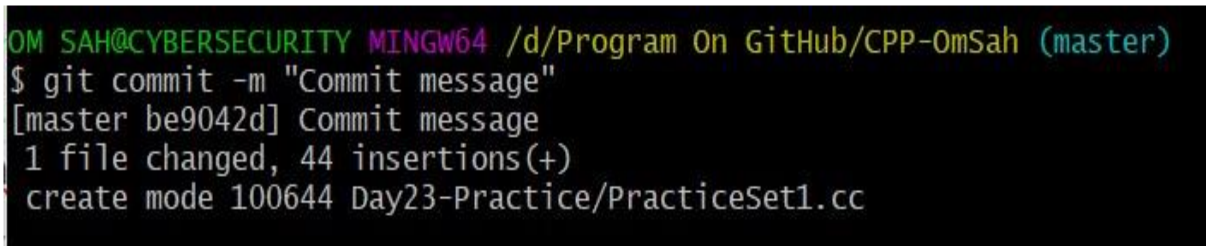


```
OM SAH@CYBERSECURITY MINGW64 /d/Program On GitHub/CPP-OmSah (master)
$ git checkout -b feature-branch
Switched to a new branch 'feature-branch'

OM SAH@CYBERSECURITY MINGW64 /d/Program On GitHub/CPP-OmSah (feature-branch)
$ git pull origin master
Warning: Permanently added the ECDSA host key for IP address '20.207.73.82' to t
he list of known hosts.
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 3), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (4/4), 753 bytes | 3.00 KiB/s, done.
From github.com:omsah84/CPP-By-OMSAH
 * branch            master     -> FETCH_HEAD
   be9042d..7dcd933  master     -> origin/master
Updating be9042d..7dcd933
Fast-forward
 Day1-Practice/SumofTwoNum.cpp | 8 ++++----
 1 file changed, 4 insertions(+), 4 deletions(-)
```

Resolve Conflicts (If Any):

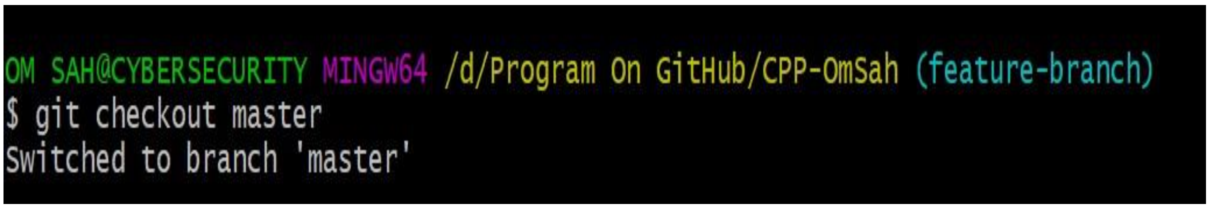
If there are conflicts between your changes and the changes in the master branch, Git will notify you. You need to resolve these conflicts manually by editing the conflicted files, marking the conflicts as resolved, and committing the changes.

Commit Your Changes (If Any Conflicts Were Resolved): After resolving conflicts, commit the changes using:



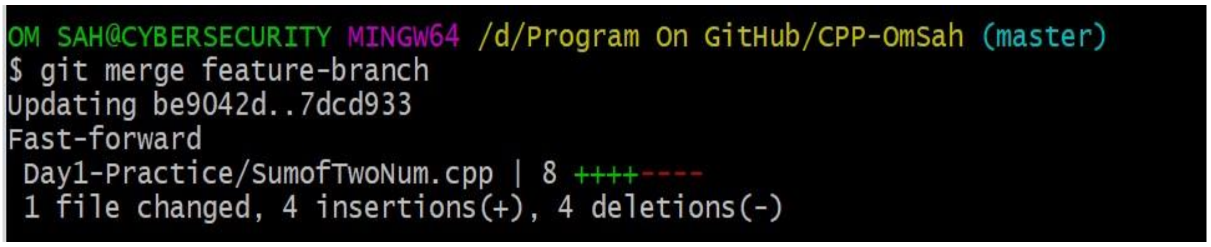
```
OM SAH@CYBERSECURITY MINGW64 /d/Program On GitHub/CPP-OmSah (master)
$ git commit -m "Commit message"
[master be9042d] Commit message
 1 file changed, 44 insertions(+)
 create mode 100644 Day23-Practice/PracticeSet1.cc
```

Switch to the Main Branch:
Return to the master branch using the git checkout command:



```
OM SAH@CYBERSECURITY MINGW64 /d/Program On GitHub/CPP-OmSah (feature-branch)
$ git checkout master
Switched to branch 'master'
```

Merge the Feature Branch:
Merge your feature branch into the main branch using the git merge command:

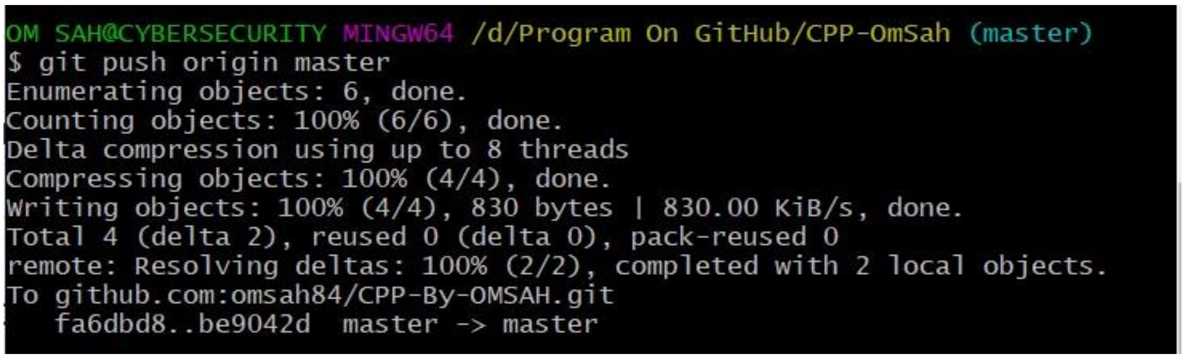


```
OM SAH@CYBERSECURITY MINGW64 /d/Program On GitHub/CPP-OmSah (master)
$ git merge feature-branch
Updating be9042d..7dcd933
Fast-forward
 Day1-Practice/SumofTwoNum.cpp | 8 ++++----
 1 file changed, 4 insertions(+), 4 deletions(-)
```

Resolve Final Conflicts (If Any):
If there are any conflicts during the merge process, repeat the conflict resolution steps mentioned earlier.

Commit the Merge:
After resolving conflicts, commit the merge changes. Git will automatically generate a merge commit message summarizing the changes.

Push the Main Branch:
Finally, push the merged changes to the remote repository's main branch:



```
OM SAH@CYBERSECURITY MINGW64 /d/Program On GitHub/CPP-OmSah (master)
$ git push origin master
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 830 bytes | 830.00 KiB/s, done.
Total 4 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:omsah84/CPP-By-OMSAH.git
   fa6dbd8..be9042d  master -> master
```

Result : Merging the branch to your Main branch has been done successfully.