Experiment No. 2

Aim: To understand version control system/ source code management, install Git, and create a GitHub account.

LO No. & Statement: (LO2): Obtain complete knowledge of the “version control system” to efficiently track change arguments with Git and GitHub.

Theory:

What is Version Control?

The purpose of version control is to allow software teams track changes to the code, while enhancing communication and collaboration between team members. Version control facilitates a continuous, simple way to develop software. Source code acts as a single source of truth and a collection of a product’s knowledge, history, and solutions. Version control (or code revision control) serves as a safety net to protect the source code from irreparable harm, giving the development team the freedom to experiment without fear of causing damage or creating code conflicts. If developers code concurrently and create incompatible changes, version control identifies the problem areas so that team members can quickly revert changes to a previous version, compare changes, or identify who committed the problem code through the revision history. With version control systems, a software team can solve an issue before progressing further into a project. Through code reviews, software teams can analyze earlier versions to understand how a solution evolved. Depending on a team's specific needs, a version control system can be local, centralized, or distributed. A local version control system stores files within a local system. Centralized version control stores changes in a single server. A distributed version control system involves cloning a Git repository.

Why use Version Control?

Software is developed to solve a user problem. Increasingly, these solutions have many different forms (e.g., mobile, embedded, SaaS) and run a variety of environments, such as cloud, on-perm, or Edge. As organizations accelerate delivery of their software solutions through DevOps, controlling and managing different versions of application artefacts - from code to configuration and from design to deployment - becomes increasingly difficult. Velocity without robust version control and traceability is like driving a car without a seatbelt. Version control facilitates coordination, sharing, and collaboration across the entire software development team. Version control software enables teams to work in distributed and asynchronous environments, manage changes and versions of code and artefacts, and resolve merge conflicts and related anomalies.

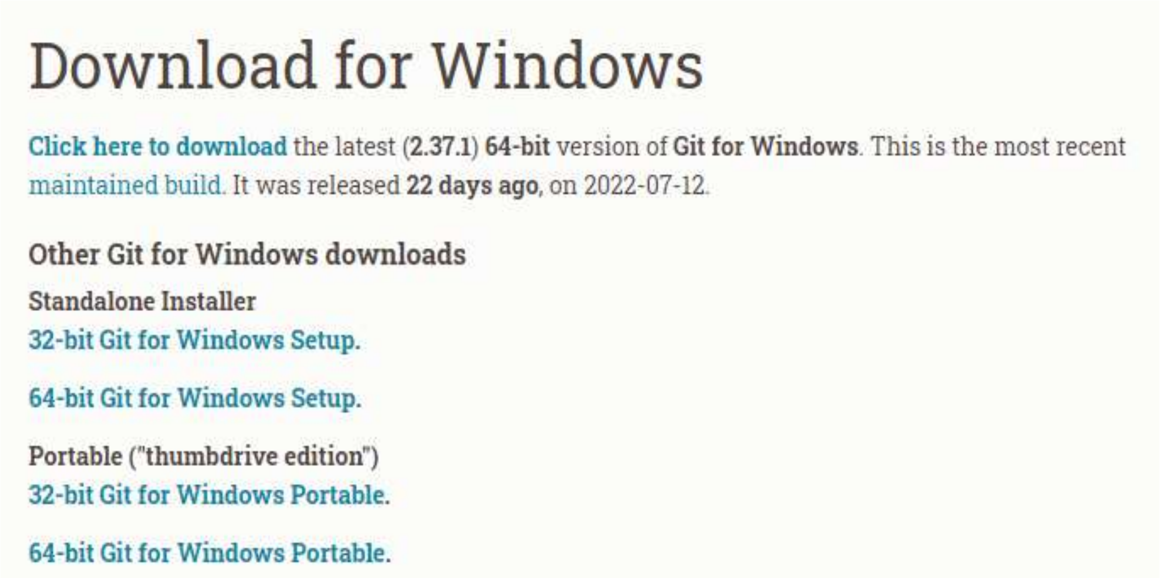
Source Code Management:

Source code management (SCM) is used to track modifications to a source code repository. SCM tracks a running history of changes to a code base and helps resolve conflicts when merging updates from multiple contributors. SCM is also synonymous with Version control.

The Importance of Source Code Management Tools:

When multiple developers are working within a shared codebase it is a common occurrence to make edits to a shared piece of code. Separate developers may be working on a seemingly isolated feature; however, this feature may use a shared code module. Therefore developer 1 working on Feature 1 could make some edits and find out later that Developer 2 working on Feature 2 has conflicting edits. Before the adoption of SCM this was a nightmare scenario. Developers would edit text files directly and move them around to remote locations using FTP or other protocols. Developer 1 would make edits and Developer 2 would unknowingly save over Developer 1's work and wipe out the changes. SCM's role as a protection mechanism against this specific scenario is known as Version Control. SCM brought version control safeguards to prevent loss of work due to conflict overwriting. These safeguards work by tracking changes from each individual developer and identifying areas of conflict and preventing overwrites. SCM will then communicate these points of conflict back to the developers so that they can safely review and address. This foundational conflict prevention mechanism has the side effect of providing passive communication for the development team. The team can then monitor and discuss the work in progress that the SCM is monitoring. The SCM tracks an entire history of changes to the code base. This allows developers to examine and review edits that may have introduced bugs or regressions.

Installation of Git:



Download for Windows

Click here to download the latest (**2.37.1**) **64-bit** version of **Git for Windows**. This is the most recent maintained build. It was released **22 days ago**, on 2022-07-12.

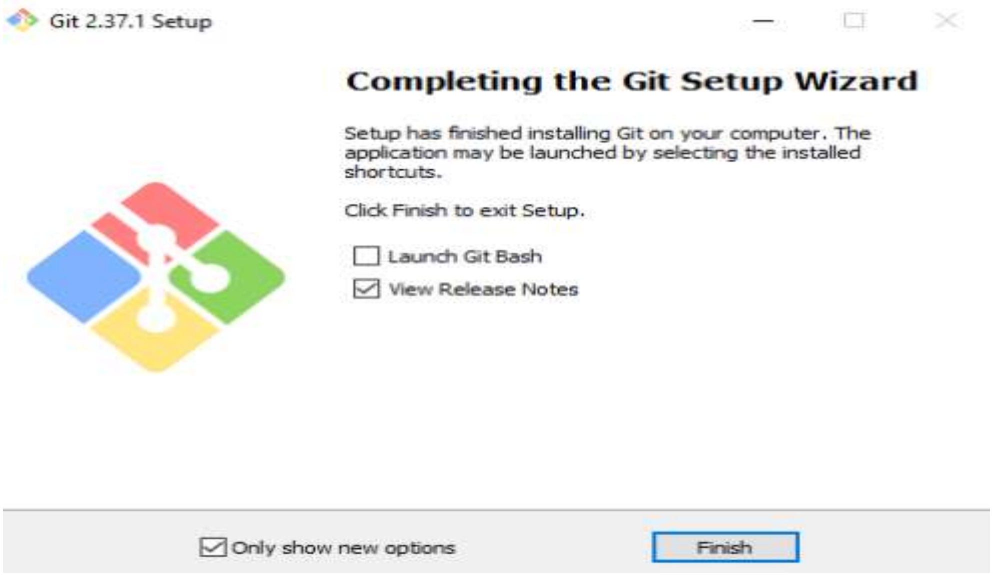
Other Git for Windows downloads

Standalone Installer
32-bit Git for Windows Setup.

64-bit Git for Windows Setup.

Portable ("thumbdrive edition")
32-bit Git for Windows Portable.

64-bit Git for Windows Portable.



Git 2.37.1 Setup

Completing the Git Setup Wizard

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

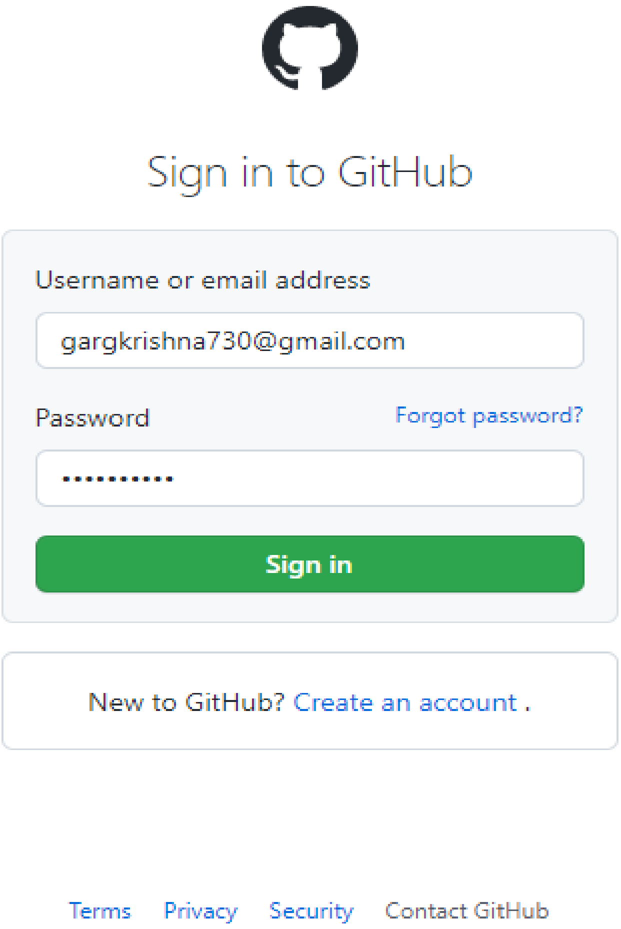
Click Finish to exit Setup.

☐ Launch Git Bash

☑ View Release Notes

☑ Only show new options

GitHub Account:



Sign in to GitHub

Username or email address

gargkrishna730@gmail.com

Password

••••••••••

New to GitHub? Create an account .

Conclusion:

In this experiment, we have learned the installation of Git and the creation of a GitHub account.

We have achieved LO2 from this experiment.

We have also achieved Program Outcomes PO1, PO2, PO3, PO4, PO5, PO12.