

## Experiment -1.1

**Student Name: Hitashi**

**Branch: CSE devops**

**Semester: 4th**

**Subject Name: Git and Github**

**UID: 22BDO10039**

**Section/Group:22BCD1-A**

**Date of Performance:17.01.24**

**Subject Code:22CSH-293**

**1. Aim/Overview of the practical:** Install Git and creating repository

**2. Task to be done:** Download Git for Mac OS. And, to make repositories.

**3. Steps for experiment/practical:**

1) Browse to the official Git website: <https://git-scm.com/downloads>.

2) Click the **download link for Mac** and allow the download to complete. Or you can simply download it from

## Downloads



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.

[View GUI Clients →](#)

### Logos

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

[View Logos →](#)

### Git via Git

If you already have Git installed, you can get the latest development version via Git itself:

```
git clone https://github.com/git/git
```

You can also always browse the current contents of the git repository using the [web interface](#).

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

## Download for macOS

There are several options for installing Git on macOS. Note that any non-source distributions are provided by third parties, and may not be up to date with the latest source release.

**Choose one of the following options for installing Git on macOS:**

### Homebrew

Install [homebrew](#) if you don't already have it, then:

```
$ brew install git
```

### MacPorts

Install [MacPorts](#) if you don't already have it, then:

```
$ sudo port install git
```

### Xcode

Apple ships a binary package of Git with [Xcode](#).

### Binary installer

Tim Harper provides an [installer](#) for Git. The latest version is [2.33.0](#), which was released over 2 years ago, on 2021-08-30.

### Building from Source

If you prefer to build from source, you can find tarballs [on kernel.org](#). The latest version is [2.43.0](#).

### Installing git-gui

If you would like to install [git-gui](#) and [gitk](#), git's commit GUI and interactive history browser, you can do so using [homebrew](#)

```
$ brew install git-gui
```



- 4) **Allow the app to make changes to your device by clicking Yes** on the User Account Control dialog that opens. Follow the installation wizard and configure Git to suit your development needs. If you are new to version control systems, the best option would be to leave the default settings.
- 5). Click **Install** and type in your password if necessary.
- 6) Confirm once again by clicking **Install Software**. With this, you have finished setting up Git on your Mac. Move on to the next step of [configuring Git](#).
- 7) Review the **GNU General Public License**, and when you're ready to install, click **Next**.

## Git for Mac Installer

The easiest way to install Git on a Mac is via the stand-alone installer:

1. Download the latest [Git for Mac installer](#).
2. Follow the prompts to install Git.
3. Open a terminal and verify the installation was successful by typing `git --version`:
4. Configure your Git username and email using the following commands, replacing Emma's name with your own. These details will be associated with any commits that you create:

```
himanshi@Himanshis-MacBook-Pro ~ % git version
himanshi@Himanshis-MacBook-Pro ~ % git config --global user.name
"Hitashikankran"
himanshi@Himanshis-MacBook-Pro ~ % git config --global user.email
"hitashi2401@gmail.com"
himanshi@Himanshis-MacBook-Pro ~ % git config --list
```

Configure  
GitHub  
Credentials

Configure your local Git installation to use your GitHub credentials by entering the following:

```
git config --global user.name "github_username"
```

```
git config --global user.email "email_address"
```

**Note:** Replace **github\_username** and **email\_address** with your GitHub credentials.

## Clone a GitHub Repository

Go to your repository on GitHub. In the top right above the list of files, open the **Clone or download** drop-down menu.

Copy the **URL for cloning over HTTPS**.

Switch to your PowerShell window, and enter the following:

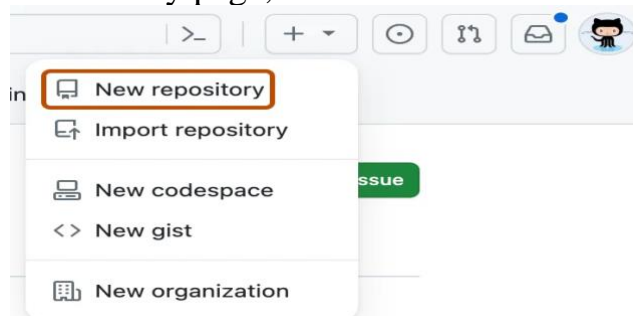
```
git clone repository_url
```

## 2.) Creating a new repository

You can create a new repository on your personal account or any organization where you have sufficient permissions.

### Creating a new repository from the web UI

1.) In the upper-right corner of any page, select + then click New repository.



2.) Optionally, to create a repository with the directory structure and files of an existing repository, select the Choose a template dropdown menu and click a template repository. You'll see template

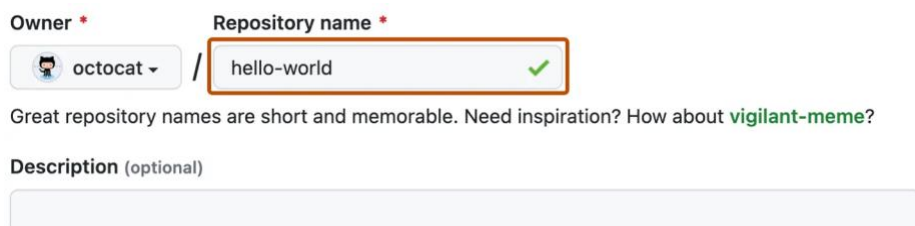
repositories that are owned by you and organizations you're a member of or that you've used before. For more information, see "[Creating a repository from a template](#)."

3.) Optionally, if you chose to use a template, to include the directory structure and files from all branches in the template, and not just the default branch, select **Include all branches**.

4.) Use the **Owner** dropdown menu to select the account you want to own the repository.



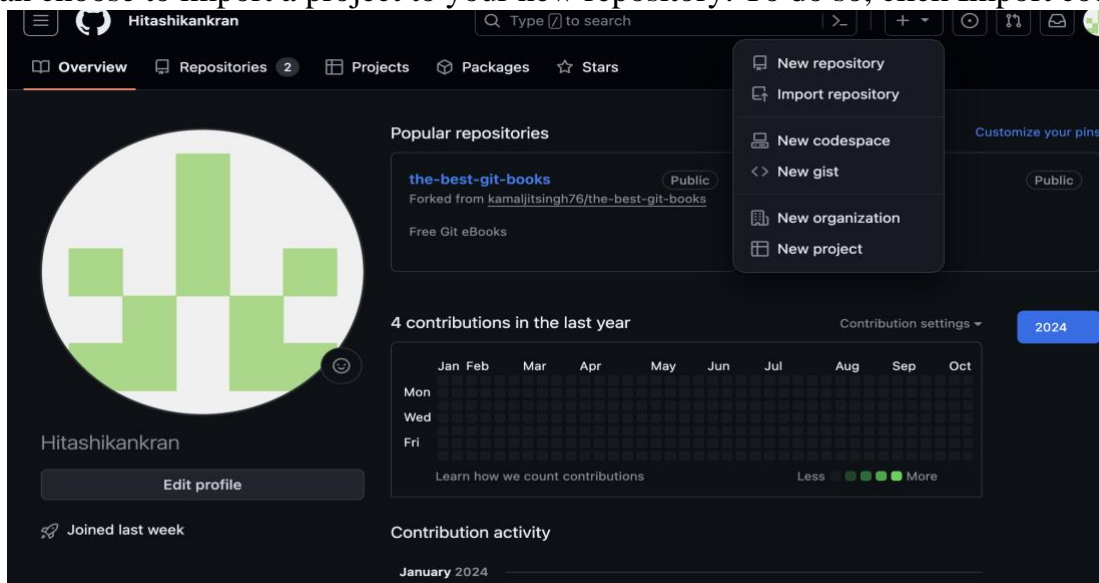
5.) Type a name for your repository, and an optional description.



6.) Choose a repository visibility. For more information, see "[About repositories](#)."

7.) If you're not using a template, there are a number of optional items you can pre-populate your repository with. If you're importing an existing repository to GitHub, don't choose any of these options, as you may introduce a merge conflict. You can add or create new files using the user interface or choose to add new files using the command line later. For more information, see "[Importing an external Git repository using the command line](#)," "[Adding a file to a repository](#)," and "[Addressing merge conflicts](#)."

- a. You can create a README, which is a document describing your project. For more information, see "[About READMEs](#)."
  - b. You can create a .gitignore file, which is a set of ignore rules. For more information, see "[Ignoring files](#)."
  - c. You can choose to add a software license for your project. For more information, see "[Licensing a repository](#)."
- 8.) Optionally, if the personal account or organization in which you're creating uses any GitHub Apps from GitHub Marketplace, select any apps you'd like to use in the repository.
- 9.) Click Create repository.
- 10.) At the bottom of the resulting Quick Setup page, under "Import code from an old repository", you can choose to import a project to your new repository. To do so, click Import code.



### 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 \* Hitashikankran / Repository name \* reposit  
reposit is available.

Great repository names are short and memorable. Need inspiration? How about [potential-octo-goggles](#)?

Description (optional)

☒ **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.](#)

① You are creating a public repository in your personal account.

**Create repository**

Hitashikankran / the-best-git-books

the-best-git-books Public  
forked from kama1jitsingh76/the-best-git-books

master 1 Branch 0 Tags

This branch is up to date with kama1jitsingh76/the-best-git-books:master. [Contribute](#) [Sync fork](#)

**Bunlong** lists books a7941eb · 6 years ago 4 Commits

File	Commit	Time
books	add some books	6 years ago
README.md	lists books	6 years ago

**README**

## The best git books

Welcome to Git eBooks collections! Enjoy learning, feel free to share with your friends!

Let star this project to keep tracking and be easy to find the project again later! Thanks!

### List of Books

Looking for a good read? Explore the books listed below and pick up one of these you will enjoy every chapter!

- Git

**About**

**Free Git eBooks**

- Readme
- Activity
- 0 stars
- 0 watching
- 63 forks

**Releases**

No releases published  
[Create a new release](#)

**Packages**

No packages published  
[Publish your first package](#)



(If Any): Image /Soft copy of graph paper to be attached here

```
Last login: Wed Jan 17 18:38:08 on ttys000
himanshi@Himanshi-MacBook-Pro ~ % git version
himanshi@Himanshi-MacBook-Pro ~ % git config --global user.name "Hitashikankran"
himanshi@Himanshi-MacBook-Pro ~ % git config --global user.email "hitashi248@gmail.com"
himanshi@Himanshi-MacBook-Pro ~ % git config --list
core.excludesfile=~/.gitignore
core.legacyheaders=false
core.quotespace=false
mergetool.keepbackup=true
push.default=simple
color.ui=auto
color.interactive=auto
repack.usedeltabaseoffset=true
alias.s=status
alias.a=git add . && git status
alias.au=git add -u . && git status
alias.am=git add . && git add -u . && git status
alias.c=commit
alias.cm=commit -m
alias.cac=commit --amend
alias.ec=git add . && git commit
alias.acm=git add . && git commit -m
alias.l=log --graph --all --pretty=format:'%C(yellow)%N%C(cyan)%N%Creset %s %C(white)- %an, %ar%Creset'
alias.lle=log --stat --abbrev-commit
alias.lgl=log --color --graph --pretty=format:'%C(bold white)%N%Creset %s %C(bold green)%N%Creset %s %C(bold blue)%N%Creset' --abbrev-commit --date=relative
alias.lll=log --color --graph --pretty=format:'%C(bold white)%N%Creset %s %C(bold blue)%N %s>%Creset %C(bold green)%N %C' --abbrev-commit
alias.d=diff
alias.m=master checkout master
alias.s=push=git push
alias.sp=push=git push
alias.al=git config --list | grep 'alias.' | sed 's/alias.\([^\s]\+\)/\1 => \2/' | sort
include.path=.gitconfig
include.path=.githubconfig
include.path=.gitcredentials
diff.xdiff.textconv=xdiff
credential.helper=ssh-keychain
user.name=Hitashikankran
user.email=hitashi248@gmail.com
himanshi@Himanshi-MacBook-Pro ~ % git clone git@github.com:Hitashikankran/repost1.git
Cloning into 'repost1'...
The authenticity of host 'github.com (20.207.73.82)' can't be established.
ED25519 key fingerprint is SHA256:DiY3wvvV6TJ3hpZlsF/zLDA82PMSVHdkz4UyCQDu.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
git@github.com: Permission denied (publickey).
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.
himanshi@Himanshi-MacBook-Pro ~ % ls
Applications (Parallels)  Downloads  Music  Public
Desktop  Library  Parallels  sql
Documents  Movies  Pictures
himanshi@Himanshi-MacBook-Pro ~ % cd repost1
cd: no such file or directory: repost1
himanshi@Himanshi-MacBook-Pro ~ % git clone https://github.com/Hitashikankran/repost1.git
Cloning into 'repost1'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
himanshi@Himanshi-MacBook-Pro ~ % sudo git clone https://github.com/Hitashikankran/repost1.git
Password:
fatal: destination path 'repost1' already exists and is not an empty directory.
himanshi@Himanshi-MacBook-Pro ~ % ls
Applications (Parallels)  Downloads  Music  Public
Desktop  Library  Parallels  repost1
Documents  Movies  Pictures  sql
himanshi@Himanshi-MacBook-Pro ~ % cd repost1
himanshi@Himanshi-MacBook-Pro ~ % git clone https://github.com/Hitashikankran/the-best-git-books.git
Cloning into 'the-best-git-books'...
remote: Enumerating objects: 27, done.
```

## Learning outcomes (What I have learnt):

1. Learnt about GitHub
2. Learnt about Git.
3. Learnt about various git commands that can be applied on Git Bash.
4. Learnt about repositories.





## DEPARTMENT OF ACADEMIC AFFAIRS

Discover. Learn. Empower.



5. Learnt about how to pull request and push.

**Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):**

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.			
2.			
3.			