1.How to check if git is available on your system

Sol-□ Open **Command Prompt (cmd)** or **PowerShell**.

□ Type the following command and press

```
 git --version
```

□ If Git is installed, you will see output like:

```
git version 2.39.1. windows.1
```

□ If Git is not installed, you will see an error message like:

```
'git' is not recognized as an internal or external command...
```

2. How to initialize a new Git repository?

Sol-

1. Open your terminal or command prompt

Navigate to the directory where you want to create the Git repository.

```
cd /path/to/your/project
```

2. Initialize the Git repository

Run the following command to create a new Git repository in that directory:

```
git init
```

This will create a hidden. git folder that tracks your project's version history.

3. How to tell git about your name and email?

Sol-

1. Set your name

```
git config --global user.name "Your Name"
```

2. Set your email

```
git config --global user. email "your.email@example.com"
```

3. Verify your settings

Check if your name and email are set correctly:

```
git config --global --list
```

This will display something like:

```
git config user.name "Your Name"
git config user. email "your.email@example.com"
```

4. How to add a file to the staging area

Sol-

1. Add a specific file

```
git add filename.txt
```

This stages filename.txt for the next commit.

2. Add multiple files

```
git add file1.txt file2.txt
```

3. Add all files in the directory

```
git add.
```

This stages all modified and new files in the current directory.

5. How to remove a file from the staging area

Sol-

1. Unstage a specific file

```
git reset filename.txt
```

This removes filename.txt from the staging area but keeps the file unchanged in your working directory.

2. Unstage all files

```
git reset
```

This removes **all staged files**, but they remain unchanged in the working directory.

3. Check the status

```
git status
```

6. How to make a commit

Sol-

1. Add files to the staging area

Before committing, you need to stage the files:

7. How to send your changes to a remote repository

Sol- To send (push) your changes to a remote repository, follow these steps:

1. Ensure you're on the correct branch

```
git branch
```

If needed, switch branches

```
git checkout main  # Or your preferred branch
```

2. Add and commit your changes

```
git add .
git commit -m "Your commit message"
```

3. Push changes to the remote repository

If your remote repository is already set up

```
git push origin main  # Replace 'main' with your branch name
```

4. If the remote is not set up yet

First, add the remote URL:

```
git remote add origin https://github.com/your-username/your-repo.git
```

Then push the changes:

```
git push -u origin main
```

```
git add filename.txt # Add a specific file
git add.           # Add all changes in the directory
```

2. Commit the changes

Use the git commit command with a message describing the changes:

```
git commit -m "Your commit message here"
```

3. Verify the commit

Check your commit history using

```
git log –online
```

7. What is the difference between clone and pull?

Sol-

Both git clone and git pull are used to interact with a remote repository, but they serve different purposes:

Command	Purpose	When to Use
git clone	Copies an entire remote repository to your local machine	When you want to download a fresh copy of a repository for the first time
git pull	Updates your local repository with the latest changes from the remote	When you already have a local repository and want to sync it with the latest changes