**Setting Up Git on Your Personal PC:**

**Install Git:**

If you haven’t already, download and install Git from the official website: Git Downloads.

**Configure Your Identity:**

Open a terminal or Git Bash and set your username and email globally using the following commands:

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

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

Replace "Your Name" and "your@email.com" with your actual name and email.

**Creating a GitHub Account:**

If you don’t have a GitHub account, sign up by navigating to GitHub and clicking “Sign up.” Follow the prompts to create your personal account and verify your email address1.

**Creating a New Repository on GitHub:**

Log in to your GitHub account.

Click the “+” icon in the top right corner and select “New repository.”

Give your repository a name, choose visibility (public or private), and skip adding a README, .gitignore, or license for now.

Sure, let's go through the steps in detail, including the explanation of what each step does, and how to check the status throughout the process.

**Step-by-Step Procedure**

1. **Create a Folder in Your Local PC**
   * **Action**: Create a folder on your PC.
   * **Example**: F:\09\_Sample\_POC\GitHub\GitRepo\01\_SampleTesting
   * **Explanation**: This folder will serve as the root directory for your local Git repository.
2. **Open the Folder in Visual Studio Code**
   * **Action**: Open Visual Studio Code, then open the folder you created.
   * **Steps**:
     + Open Visual Studio Code.
     + Click on File -> Open Folder.
     + Navigate to F:\09\_Sample\_POC\GitHub\GitRepo\01\_SampleTesting and select it.
3. **Open Terminal in Visual Studio Code**
   * **Action**: Open the terminal inside Visual Studio Code.
   * **Steps**:
     + Go to View -> Terminal or press Ctrl + (backtick) or Cmd + (backtick) on Mac.
4. **Clone the GitHub Repository**
   * **Action**: Clone the remote repository into your local folder.
   * **Command**:

git clone https://github.com/Thillaishanmugam1998/01\_SampleTesting.git

* + **Explanation**: This command will copy the remote repository's content into a new folder named 01\_SampleTesting inside F:\09\_Sample\_POC\GitHub\GitRepo.

1. **Check Cloned Repository in Local Folder**
   * **Action**: Verify that the repository has been cloned successfully.
   * **Explanation**: Navigate to F:\09\_Sample\_POC\GitHub\GitRepo\01\_SampleTesting in your file explorer. You should see the contents of the repository, including any files that were in the remote repo.
2. **Add a New File**
   * **Action**: Create a new file in the cloned repository.
   * **Example**: Create a file named newfile.py.
   * **Explanation**: The new file will appear with a green color 'U' symbol in Visual Studio Code, indicating it is untracked.

**Git Commands and Checking Status**

1. **Stage the New File**
   * **Action**: Stage the new file to be committed.
   * **Command**:

git add newfile.py

* + **Explanation**: This command tells Git to start tracking the new file. It changes the status from untracked to staged.

1. **Check Status**
   * **Action**: Check the status of your working directory.
   * **Command**:

git status

* + **Explanation**: This command shows the current status of your working directory, including staged, unstaged, and untracked files.

1. **Commit the Changes**
   * **Action**: Commit the staged file with a descriptive message.
   * **Command**:

git commit -m "Added newfile.py"

* + **Explanation**: This command records the changes in the repository with a message describing what was done.

1. **Check Status Again**
   * **Action**: Check the status again to ensure there are no pending changes.
   * **Command**:

git status

* + **Explanation**: This should show a clean working directory if everything is committed.

1. **Push the Changes to GitHub**
   * **Action**: Push the committed changes to the remote repository.
   * **Command**:

git push origin main

* + **Explanation**: This command sends the committed changes to the remote repository. Replace main with master if your default branch is named master.

To push the new changes (extra two lines) to your GitHub repository, follow these steps:

1. **Modify the File**
   * **Action**: Open newfile.py in Visual Studio Code and add the extra two lines of code.
   * **Explanation**: Make your changes and save the file. The file will now be modified.
2. **Check the Status**
   * **Action**: Open the terminal in Visual Studio Code and check the status.
   * **Command**:

git status

* + **Explanation**: This will show that newfile.py has been modified.

1. **Stage the Modified File**
   * **Action**: Stage the modified file for commit.
   * **Command**:

git add newfile.py

* + **Explanation**: This stages the changes to newfile.py, preparing them for commit.

1. **Check the Status Again**
   * **Action**: Verify that the file has been staged.
   * **Command**:

git status

* + **Explanation**: This should show newfile.py in the "Changes to be committed" section.

1. **Commit the Changes**
   * **Action**: Commit the staged changes with a descriptive message.
   * **Command**:

git commit -m "Updated newfile.py with additional lines"

* + **Explanation**: This command records the changes in the repository with a message describing what was done.

1. **Push the Changes to GitHub**
   * **Action**: Push the committed changes to the remote repository.
   * **Command**:

git push origin main

* + **Explanation**: This command sends the committed changes to the remote repository. Replace main with master if your default branch is named master.