**Task: GitHub Repository Setup and Git Operations**

**1. Establish a New Directory and Populate It with Script Files**

1. Open your terminal or command prompt.
2. Create a new directory:
3. mkdir Git-locale
4. cd Git-locale
5. Add script files (for example, a Python script):
6. echo "print('Hello, GitHub!')" > hello.py

A screenshot of a computer

AI-generated content may be incorrect.

**2. Initiate an Empty Repository on GitHub**

1. Go to [GitHub](https://github.com/) and log in.
2. Click on the **+** icon in the top right corner → **New repository**.
3. Enter a repository name, Learnings.
4. Select **Public**, and don’t initialize with a README (we’ll push that from local).
5. Click **Create repository**.

A screenshot of a computer

AI-generated content may be incorrect.

**3. Convert Local Directory into a Git Repository and Push to GitHub**

# Initialize git in your local directory

git init

A computer screen with text

AI-generated content may be incorrect.

# Add the files

git add .

A computer screen with white text

AI-generated content may be incorrect.

# Commit the changes

git commit -m "Initial commit"

A computer screen with white text

AI-generated content may be incorrect.

# Link to the GitHub repository (replace YOUR-USERNAME and REPO-NAME accordingly)

git remote add origin https://github.com/Surya-pkh/Learnigs.git

# Push to GitHub

git push -u origin master

A computer screen with white text

AI-generated content may be incorrect.

**4. Perform Git Merge, Rebase, and Stash Commands**

**A. Git Merge**

1. Create a new branch and switch to it:

git checkout -b branch1

A computer screen with blue text

AI-generated content may be incorrect.

1. Make a change in hello.py, then commit it.

echo "print('This is a new feature')" >> hello.py

1. git add hello.py

git commit -m "Add new feature"

A computer screen with white and blue text

AI-generated content may be incorrect.

1. Switch back to the master branch and merge:

git checkout master

1. git merge branch1

A computer screen with white text

AI-generated content may be incorrect.

**B. Git Rebase**

1. Create another branch and make a different change:

git checkout -b branch2

echo "# This line is added for rebase demo" >> hello.py

git add hello.py

git commit -m "Change for rebase"

1. Rebase it onto master:

git checkout master

git checkout branch2

git rebase master

A screenshot of a computer program

AI-generated content may be incorrect.

**C. Git Stash**

1. Make uncommitted changes:
2. echo "# Temporary change" >> hello.py
3. Stash them:

git stash

1. Your working directory is clean. To apply them back:

git stash pop

A screenshot of a computer program

AI-generated content may be incorrect.