## Git Reset using Visual Studio

**Step 1: Create a GitHub Repository**

1. **Go to GitHub**: Open [GitHub](https://github.com) and log in to your account.
2. **Create New Repository**:
   * Click the "+" icon in the upper-right corner and select "New repository."
   * Enter a repository name (e.g., reset-demo), choose the visibility (Public/Private), and click "Create repository."
3. **Get the Repository URL**: Copy the repository’s URL from the page that appears (e.g., https://github.com/username/reset-demo.git).

**Step 2: Clone the Repository in Visual Studio**

1. **Open Visual Studio**: Launch Visual Studio.
2. **Clone the Repository**:
   * Go to File > Clone Repository.
   * Paste the copied GitHub repository URL and click "Clone."

**Step 3: Create and Edit Files in Visual Studio**

1. **Add New Files**:
   * In the Solution Explorer, right-click on your repository folder and select Add > New Item.
   * Add 2–3 new files (e.g., file1.txt, file2.txt) with some sample content.
2. **Make Changes (1st Commit)**:
   * Open file1.txt, add some content (e.g., This is file1 content), and save the file.
   * Go to the Git Changes window (View > Git Changes).
   * Add a commit message like "Added content to file1" and click Commit All.
3. **Make Changes (2nd Commit)**:
   * Open file2.txt, add content (e.g., This is file2 content), and save the file.
   * Commit the changes with a new message like "Added content to file2."
4. **Make Changes (3rd Commit)**:
   * Open file1.txt again, modify its content (e.g., Updated content in file1), and save the file.
   * Commit the changes with a new message like "Updated file1."

**Step 4: Use git reset in Visual Studio**

**Option 1: Soft Reset (Keep Changes in Files)**

1. **View Commit History**:
   * Go to the Git Repository window (View > Git Repository).
   * Right-click the repository in the left panel and select View History to open the commit history.
2. **Reset the Last Commit**:
   * Right-click on the commit **before** the last one you made (for example, the second commit) and select Reset -> Reset Soft.
   * This will move the HEAD pointer to that commit but keep the changes you made in the last commit in your working directory (they are unstaged).
3. **Check Files**:
   * Go to the Git Changes window, and you'll see the changes from your last commit unstaged and available to be committed again.

**Option 2: Hard Reset (Discard Changes)**

1. **View Commit History**:
   * In the Git Repository window, right-click the repository and select View History again.
2. **Hard Reset to a Previous Commit**:
   * Right-click the second commit (or whichever one you want to revert to) and select Reset -> Reset Hard.
   * This will move the HEAD pointer to that commit and discard any changes made after that commit, effectively undoing them.
3. **Verify**:
   * After the reset, the repository will look as it did at the point of the commit you reset to. All changes from subsequent commits will be lost.

**Step 5: Push Changes to GitHub**

1. **Push the Changes**:
   * Once you have performed the reset and made any further changes, you can push the changes back to GitHub by going to the Git Changes window and clicking on Push.

**Note**: If you performed a hard reset or a soft reset and want to overwrite the history in GitHub, you might need to force push (Push (Force)) to rewrite the remote history.