1. **Initialize the Repository and Amending a Commit:**

* Using ‘git init’ command you can initialize a new git repository in that directory.

Here maybe you will get an error like “The term 'git' is not recognized as a name of a cmdlet, function, script file, or executable program.” To solve this problem, we must add Git to PATH

[Environment]::SetEnvironmentVariable("Path", $env:Path + ";C:\Program Files\Git\bin", [EnvironmentVariableTarget]::User)

Verify by - $env:Path -split ';'

Try to check "git --version"

A computer screen shot of a black screen

Description automatically generated

* Create a text file named “example.txt” with some text in it (“Initial Content”).

"Initial Content" | Out-File -FilePath "./example.txt"

A screenshot of a computer

Description automatically generated

* We can stage the file by using ‘git add example.txt’ command. You can see on right side that the text file moved in to staged changes tab.

A computer screen with white text

Description automatically generated

* To commit this file, use this command. You can see the changes on right side pane.

‘git commit -m “Initial commit with example.txt”

A computer screen with a black background

Description automatically generated

* To modify the content of the example.txt file and amend the previous commit without changing message. Use following command:

“Modify Content” | Out-File -FilePath “./example.txt” you can see code ‘M’ beside the file.

A computer screen with many colorful text

Description automatically generated

First, add the file in git. To amend the file: git commit --amend --no-edit here ‘—no-edit’ means no changes in the comment.

A computer screen with white text

Description automatically generated

1. **Changing a Commit Message:**

* To update commit message use: git commit –amend -m “Update commit message”

**A screenshot of a computer screen

Description automatically generated**

1. **Unstaging Changes from a Commit:**

* Create two files using name file1.txt and file2.txt with some content in it

"Content for file 1" | Out-File -FilePath "./file1.txt"

"Content for file 2" | Out-File -FilePath "./file2.txt"

A computer screen with a black background

Description automatically generated

* Stage both files: you will see code ‘A’ next to these files.

git add file1.txt

git add file2.txt

git add . (to all files)

A screenshot of a computer

Description automatically generated

* Committing the files:

git commit -m "Added file1.txt and file2.txt"

A computer screen shot of a black screen

Description automatically generated