



Week 1: Coding Assignment

URL to GitHub Repository:

<https://github.com/smiller3033/week-1-assignment>

URL to Your Coding Assignment Video:

<https://www.youtube.com/watch?v=58XaPUL7SVY>

Instructions:

- In Visual Studio Code, write the code that accomplishes the objectives listed below and ensures that the code compiles and runs as directed.
- Create a new repository on GitHub for this week's assignments and push this document, with your project code, to the repository.
- Include the URLs for this week's repository and video where instructed.
- Submit this document as a .PDF file in the LMS.

Assignment Steps:

- The link below has a zipped file that contains an empty directory (folder) for your assignments.
- Download the file to your computer and unzip it.
- This directory (folder) should be used to organize each week's projects in the course.

https://drive.google.com/file/d/1WDc_WJ8I0MfwbrbmtMsxHdTpupZsPjXT/view

- Following the Git/GitHub tutorial in your week 0 video:
 - Create a directory (folder) inside **Week 01 - CLI, Source Control, and Variables**
 - Create a repository on the GitHub website.
 - **Push** your directory of files to GitHub as instructed in the video.
 - After your first push, please ensure that you make some changes to your directory (folder), such as adding a new file or changing your code.
 - **Push** those changes to your repository a second time (as shown in the video).

<https://www.youtube.com/watch?v=NGeksLUB1e8>

- When complete, paste a screenshot of your terminal or command prompt that shows.



Week 1: Coding Assignment

Video Steps:

- Create a video, up to five minutes max, showing and explaining how your project works with an emphasis on the portions you contributed.
- This video should be done using screen share and voice over.
- This can easily be done using Zoom, although you don't have to use Zoom, it's just what we recommend.
 - You can create a new meeting, start screen sharing, and start recording.
 - This will create a video recording on your computer.
- This should then be uploaded to a publicly accessible site, such as YouTube.
 - Ensure the link you share is **PUBLIC** or **UNLISTED**!
 - If it is not accessible by your grader, your project will be graded based on what they can access.

```

Last login: Tue Oct 4 11:10:01 on ttys000
stephenmiller@Stephens-MacBook-Pro ~ % cd /Users/stephenmiller/Library/CloudStorage/OneDrive-UMassLowell/Class\ Assignments/Front\ End\ Bootcamp/Promineo_Tech/Week-01-CLI_Source_Control_and_Variables
stephenmiller@Stephens-MacBook-Pro Week-01-CLI_Source_Control_and_Variables % mkdir test-repo
stephenmiller@Stephens-MacBook-Pro Week-01-CLI_Source_Control_and_Variables % cd test-repo
stephenmiller@Stephens-MacBook-Pro test-repo % touch README.md
stephenmiller@Stephens-MacBook-Pro test-repo % git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /Users/stephenmiller/Library/CloudStorage/OneDrive-UMassLowell/Class Assignments/Front End Bootcamp/Promineo_Tech/Week-01-CLI_Source_Control_and_Variables/test-repo/.git/
stephenmiller@Stephens-MacBook-Pro test-repo % git status
On branch master

```

No commits yet

Untracked files:

```

(use "git add <file>..." to include in what will be committed)
    README.md

```

nothing added to commit but untracked files present (use "git add" to track)

```

stephenmiller@Stephens-MacBook-Pro test-repo % git add .
stephenmiller@Stephens-MacBook-Pro test-repo % git status
On branch master

```

No commits yet

Changes to be committed:

```

(use "git rm --cached <file>..." to unstage)
    new file:   README.md

```

```

stephenmiller@Stephens-MacBook-Pro test-repo % git commit -m "first commit"
[master (root-commit) 6376c24] first commit
Committer: Stephen Miller <stephenmiller@Stephens-MacBook-Pro.local>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

```

```
git config --global --edit
```

After doing this, you may fix the identity used for this commit with:

```
git commit --amend --reset-author
```

```

1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 README.md
stephenmiller@Stephens-MacBook-Pro test-repo % git branch -M main
stephenmiller@Stephens-MacBook-Pro test-repo %
stephenmiller@Stephens-MacBook-Pro test-repo % git remote add origin https://github.com/smiller3033/week-1-assignment.git
stephenmiller@Stephens-MacBook-Pro test-repo % git push -u origin main
Username for 'https://github.com': smiller3033
Password for 'https://smiller3033@github.com':
remote: Support for password authentication was removed on August 13, 2021.
remote: Please see https://docs.github.com/en/get-started/getting-started-with-git/about-remote-repositories#cloning-with-https-urls for in
formation on currently recommended modes of authentication.
fatal: Authentication failed for 'https://github.com/smiller3033/week-1-assignment.git/'
stephenmiller@Stephens-MacBook-Pro test-repo % git push -u origin main
Username for 'https://github.com': smiller3033
Password for 'https://smiller3033@github.com':
remote: Support for password authentication was removed on August 13, 2021.
remote: Please see https://docs.github.com/en/get-started/getting-started-with-git/about-remote-repositories#cloning-with-https-urls for in
formation on currently recommended modes of authentication.
fatal: Authentication failed for 'https://github.com/smiller3033/week-1-assignment.git/'
stephenmiller@Stephens-MacBook-Pro test-repo % brew tap microsoft/git
Running 'brew update --auto-update'...
==> Auto-updated Homebrew!
Updated 1 tap (homebrew/core).
==> New Formulae
edenccommon      mycorrhiza      netcdf-fortran   sambamba
jj               netcdf-cxx      rome             tbls

```

You have 1 outdated formula installed.
You can upgrade it with **brew upgrade**
or list it with **brew outdated**.

==> Tapping microsoft/git

```

Cloning into '/usr/local/Homebrew/Library/Taps/microsoft/homebrew-git'...
remote: Enumerating objects: 388, done.
remote: Counting objects: 100% (259/259), done.
remote: Compressing objects: 100% (174/174), done.
remote: Total 388 (delta 144), reused 150 (delta 85), pack-reused 129
Receiving objects: 100% (388/388), 92.70 KiB | 1.78 MiB/s, done.

```

```

Resolving deltas: 100% (219/219), done.
Tapped 4 casks (20 files, 120KB).
stephenmiller@Stephens-MacBook-Pro test-repo % brew install --cask git-credential-manager-core
==> Downloading https://github.com/GitCredentialManager/git-credential-manager/r
==> Downloading from https://objects.githubusercontent.com/github-production-rel
##### 100.0%
==> Installing Cask git-credential-manager-core
==> Running installer for git-credential-manager-core; your password may be nece
Package installers may write to any location; options such as '--appdir' are ignored.
Password:
installer: Package name is Git Credential Manager
installer: Installing at base path /
installer: The install was successful.
📦 git-credential-manager-core was successfully installed!
stephenmiller@Stephens-MacBook-Pro test-repo % git push -u origin main
info: please complete authentication in your browser...
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 227 bytes | 227.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/smiller3033/week-1-assignment.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
stephenmiller@Stephens-MacBook-Pro test-repo % touch index.html
stephenmiller@Stephens-MacBook-Pro test-repo % touch script.js
stephenmiller@Stephens-MacBook-Pro test-repo % git add .
stephenmiller@Stephens-MacBook-Pro test-repo % git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   index.html
        new file:   script.js

stephenmiller@Stephens-MacBook-Pro test-repo % git push -u origin main
branch 'main' set up to track 'origin/main'.
Everything up-to-date
stephenmiller@Stephens-MacBook-Pro test-repo % git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   index.html
        new file:   script.js

stephenmiller@Stephens-MacBook-Pro test-repo % git commit "added new html and js files"
error: pathspec 'added new html and js files' did not match any file(s) known to git
stephenmiller@Stephens-MacBook-Pro test-repo % git commit -m "added new html and js files"
[main 49feca7] added new html and js files
Committer: Stephen Miller <stephenmiller@Stephens-MacBook-Pro.local>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly:

    git config --global user.name "Your Name"
    git config --global user.email you@example.com

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 index.html
create mode 100644 script.js
stephenmiller@Stephens-MacBook-Pro test-repo % git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 287 bytes | 287.00 KiB/s, done.
Total 2 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/smiller3033/week-1-assignment.git
 6376c24..49feca7  main -> main
branch 'main' set up to track 'origin/main'.
stephenmiller@Stephens-MacBook-Pro test-repo % git config --global user.name "Stephen Miller"
stephenmiller@Stephens-MacBook-Pro test-repo % git config --global user.email thewallmiller@me.com
stephenmiller@Stephens-MacBook-Pro test-repo % git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
stephenmiller@Stephens-MacBook-Pro test-repo % git branch
* main
stephenmiller@Stephens-MacBook-Pro test-repo %

```