

CSP2101 Scripting Languages: Workshop 12

Aim:

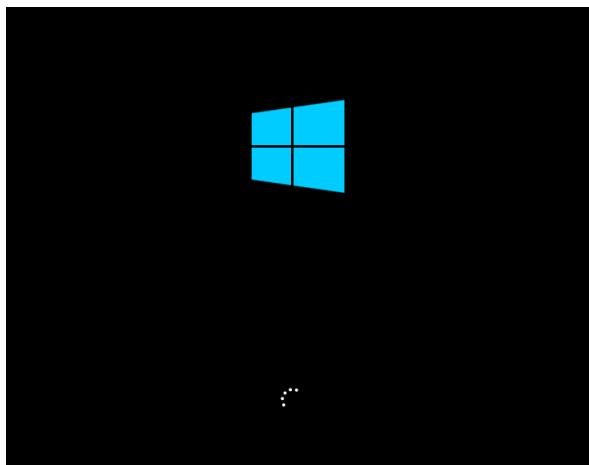
To introduce you to Windows Scripting in Batch and PowerShell

1. Use a Windows Environment
2. Use Git for Windows
3. Navigate and use Git with the cmd prompt
4. Writing some simple Batch scripts
5. Writing a simple PowerShell script

Task 1: Use a Windows Environment

For this week's work we'll be taking a break from Unix-like systems and writing some scripts for Windows.

1. To do that, you will need Administrator access to a Microsoft Windows Operating System.
2. If you do not have a Windows computer, or you are using a restricted Lab computer, you can download a free Windows virtual machine from Microsoft at the following link:
<https://developer.microsoft.com/en-us/microsoft-edge/tools/vms/>



(Windows 10 Start-up screen)

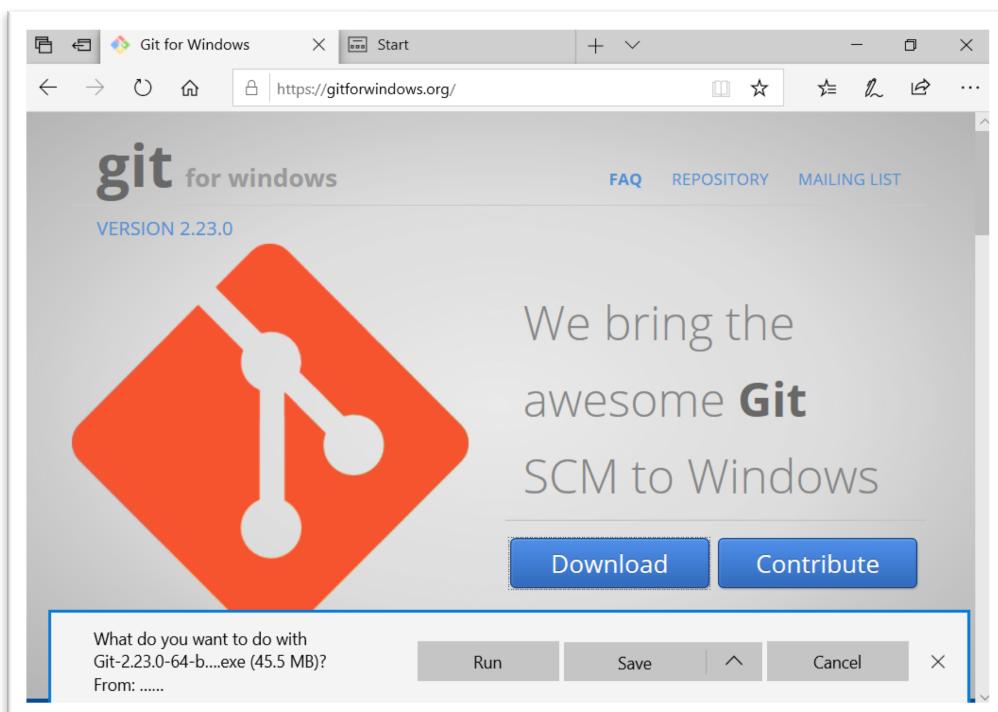
3. You also need to install "Visual Studio Code" Text editor at the following link:
<https://code.visualstudio.com/>

Task 2: Use Git for Windows

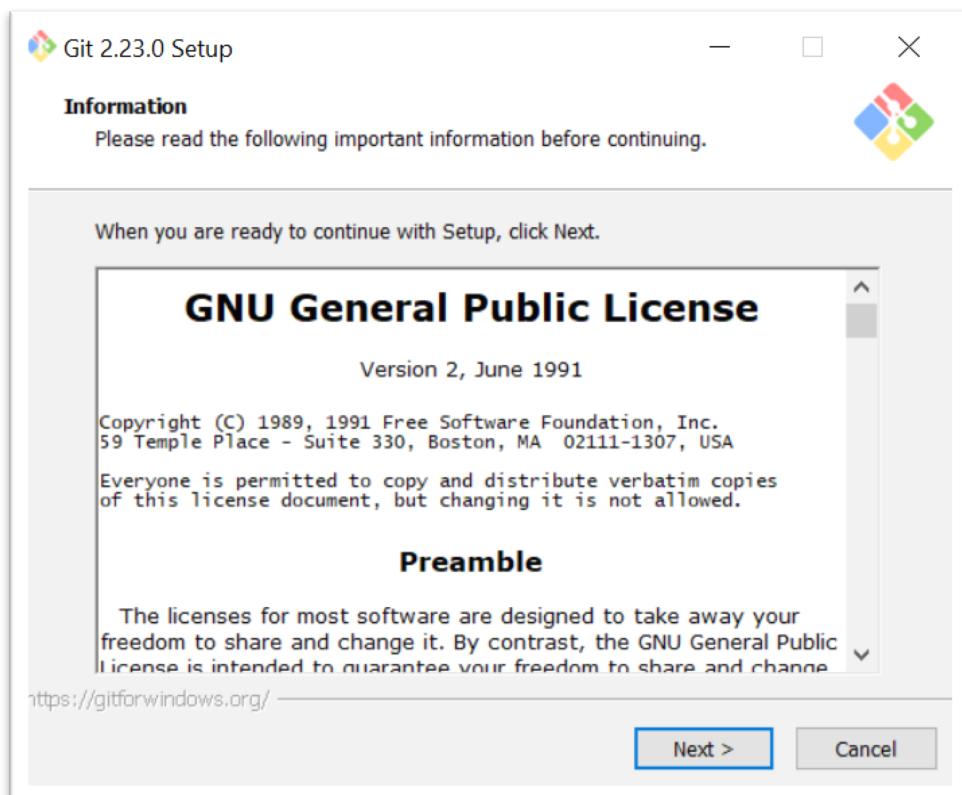
1. If your Windows environment does not yet have git installed, you should install it from
<https://gitforwindows.org>
2. This should allow us to clone the repository from our previous week's work into our windows system. Git for Windows also includes a bash shell that can be used to run git commands from the command line.

How to install:

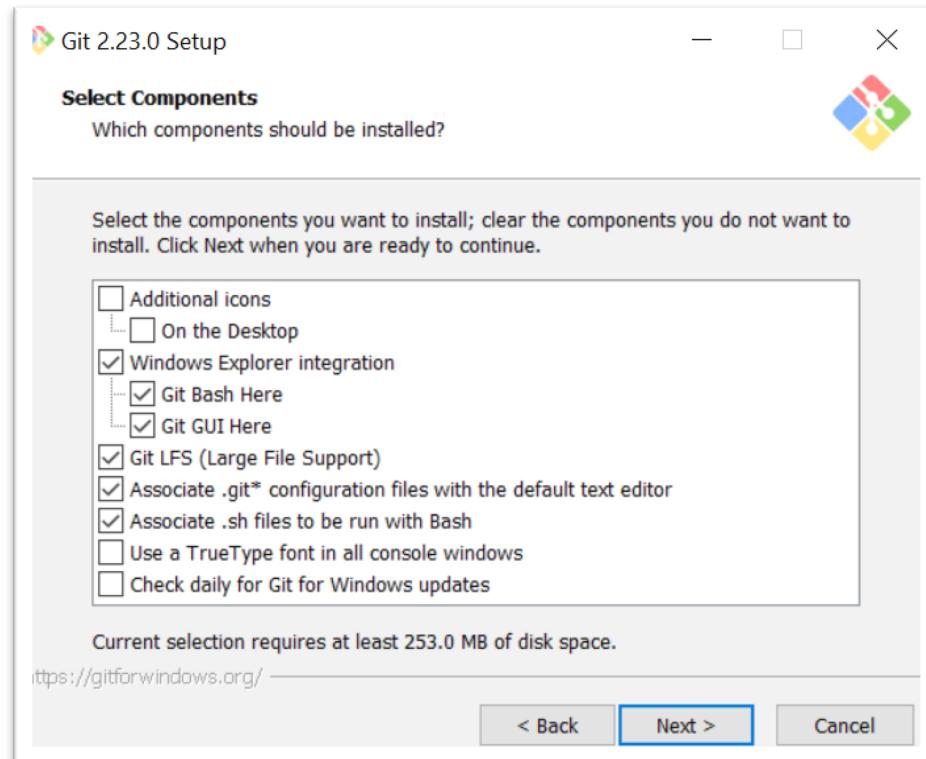
1. Go to the git for windows website and download the installer.



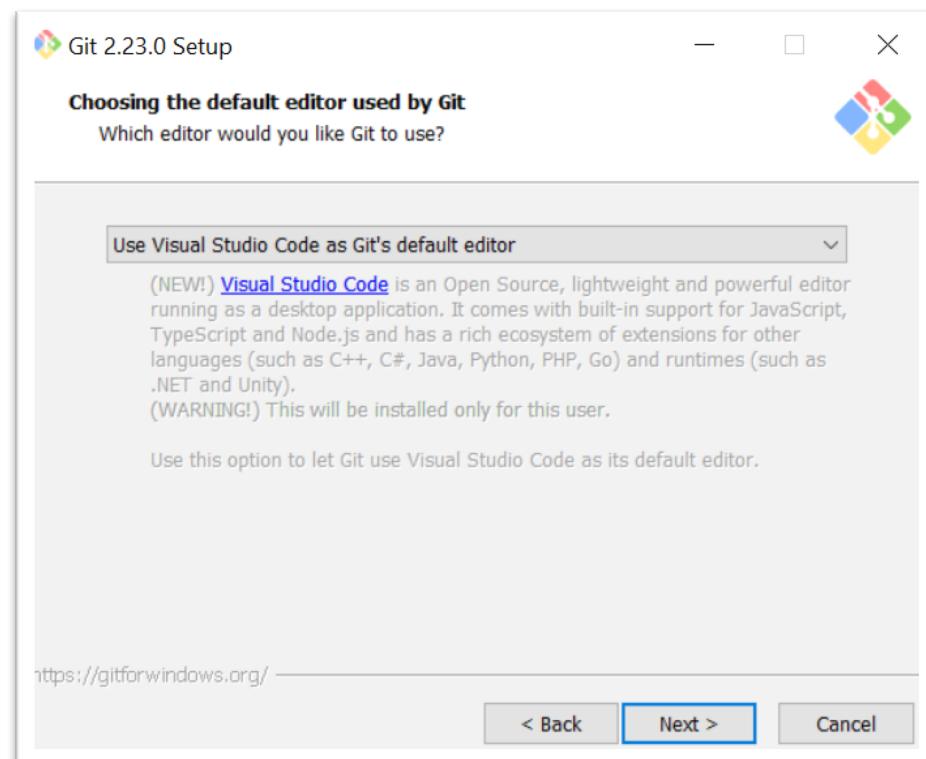
2. Follow the instructions in the "Setup Wizard"



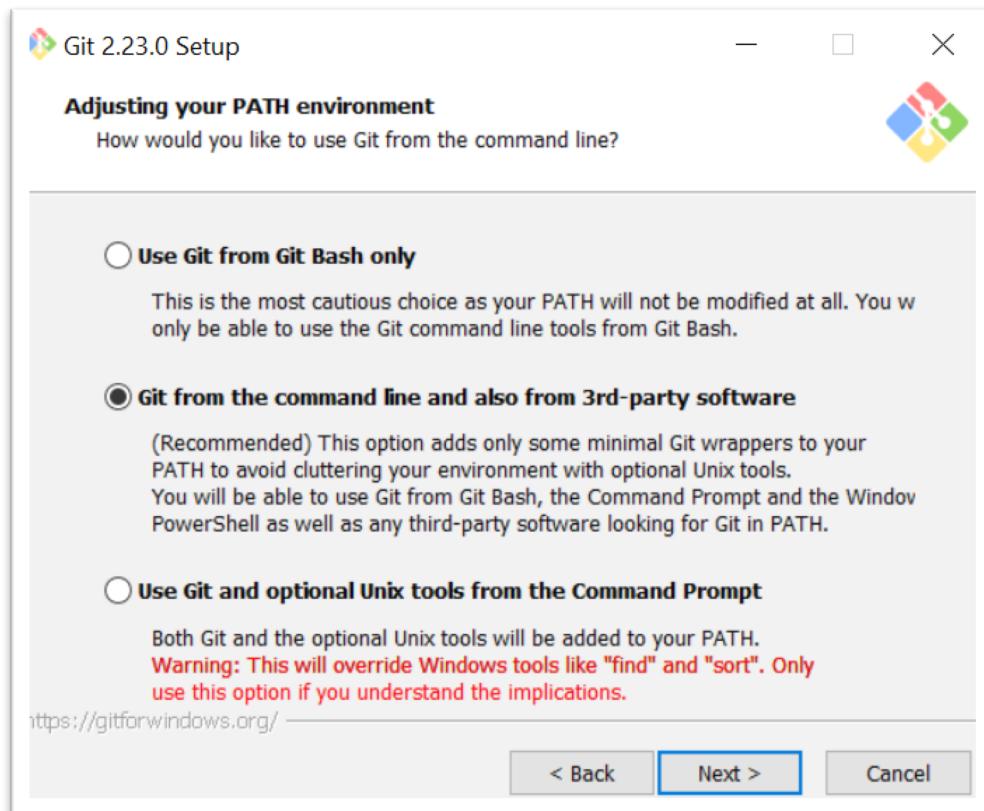
3. When you reach the “Select Components” section, select the components you wish to install:



4. When you reach the “Choose Default Editor” section, Select your preferred text editor for use with git (Note: VSCode may need to be installed first)



5. Make sure that “git from the command line and also 3rd party software” is the selected option

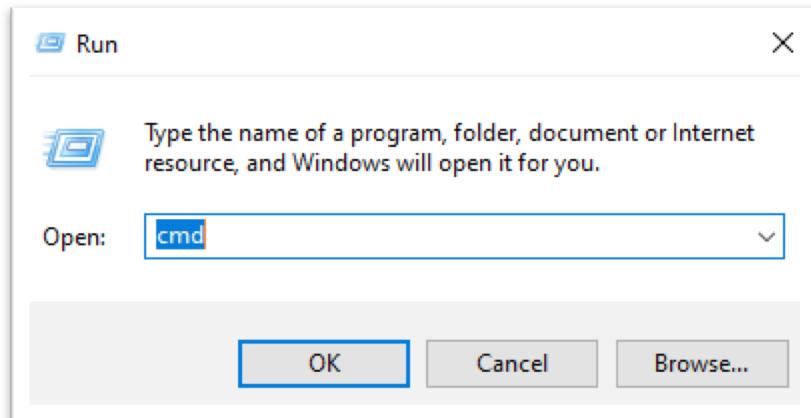


6. Continue to click next until the installation is complete

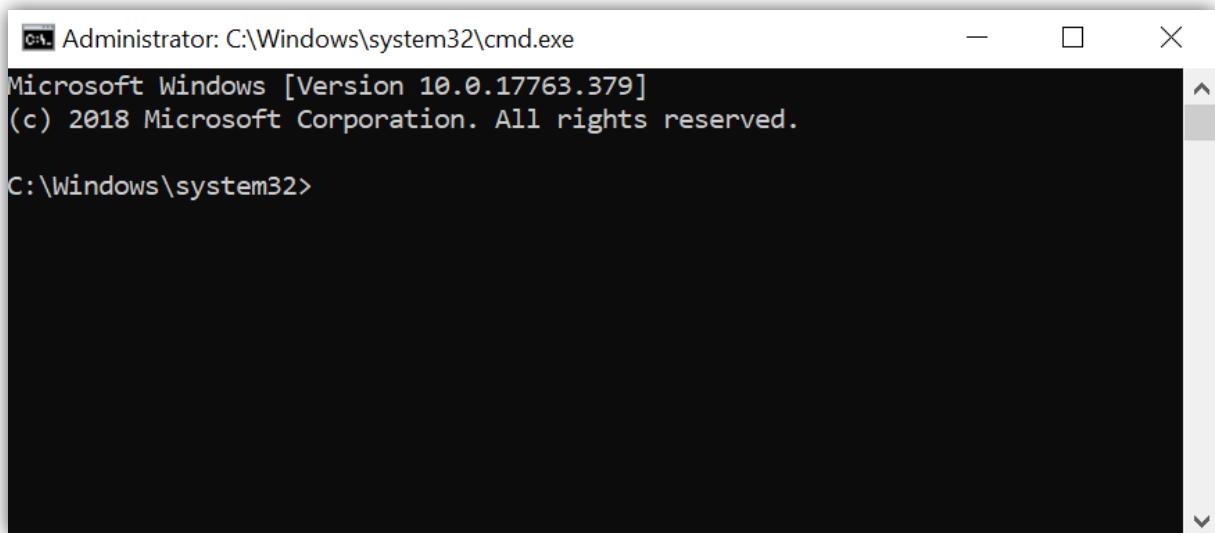
Task 3: Navigate and use Git with the cmd prompt

Let's explore the windows command line interface.

1. Press the “Win+R” on your keyboard to open the “Run” dialog box



2. Type “cmd” and then hit “ctrl+shift+enter” to open an administrator command prompt:

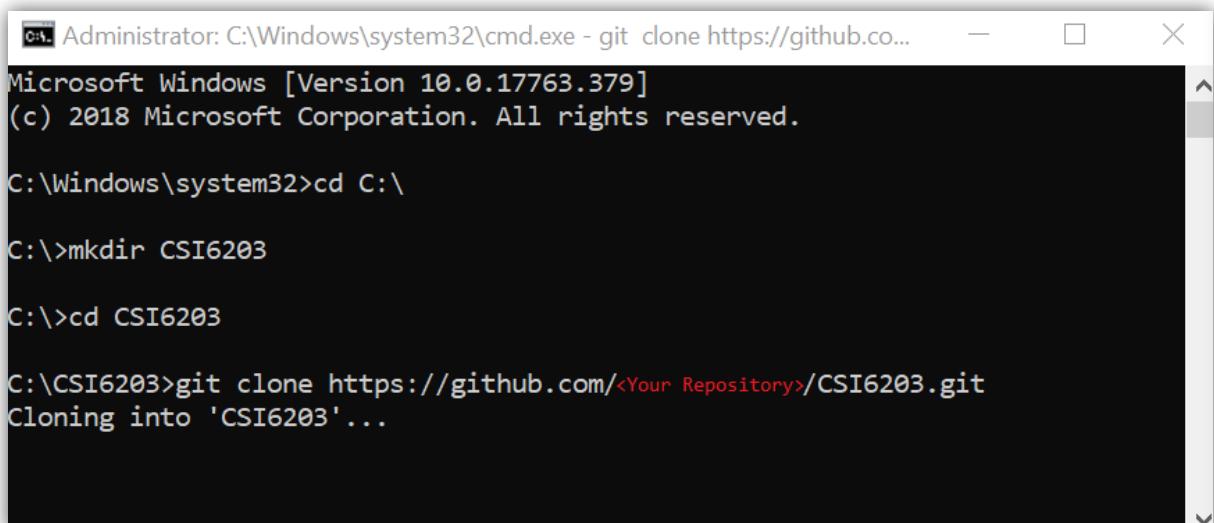


3. Navigate to the C:\ drive and create a new folder named CSI6203

Example commands:

```
C:\Windows\system32>cd C:\  
C:\>mkdir CSI6203  
C:\>cd CSI6203  
C:\CSI6203>
```

4. Next let's download our work using the git clone command:



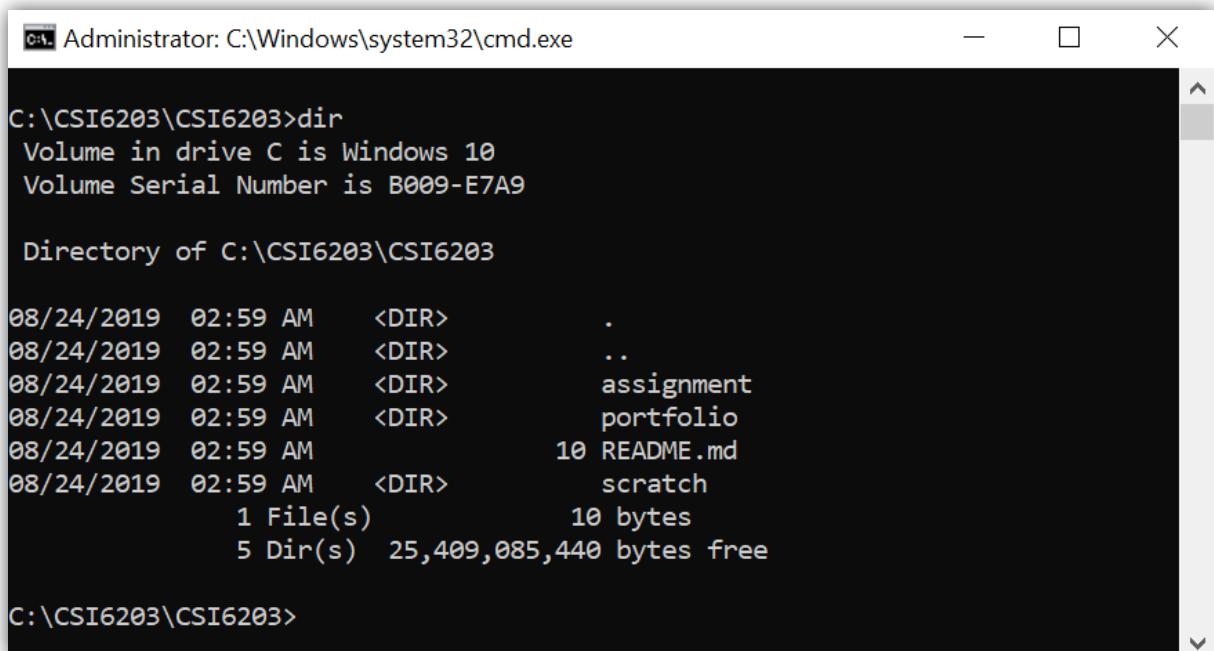
```
Administrator: C:\Windows\system32\cmd.exe - git clone https://github.co...
Microsoft Windows [Version 10.0.17763.379]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>cd C:\
C:>mkdir CSI6203
C:>cd CSI6203
C:\CSI6203>git clone https://github.com/<Your Repository>/CSI6203.git
Cloning into 'CSI6203'...
```

Note: At this point, you will be prompted to input your git credentials.

5. You should be able to navigate through the folder structure just like in bash.

- a. cmd uses the "dir" command instead the "ls" command in bash.
- b. Try to navigate through your downloaded git repository and make sure it's all there:



```
Administrator: C:\Windows\system32\cmd.exe
C:\CSI6203\CSI6203>dir
Volume in drive C is Windows 10
Volume Serial Number is B009-E7A9

Directory of C:\CSI6203\CSI6203

08/24/2019  02:59 AM    <DIR>        .
08/24/2019  02:59 AM    <DIR>        ..
08/24/2019  02:59 AM    <DIR>        assignment
08/24/2019  02:59 AM    <DIR>        portfolio
08/24/2019  02:59 AM            10 README.md
08/24/2019  02:59 AM    <DIR>        scratch
                           1 File(s)      10 bytes
                           5 Dir(s)   25,409,085,440 bytes free

C:\CSI6203\CSI6203>
```

6. Now that we have our work, we can open this folder in Visual Studio Code the same way as we did in bash, by typing:

```
code .
```

Task 4: Writing some simple Batch scripts

1. In the “Week 11” folder, write a new Batch script named “hello.bat” and write the following contents into it:

```
@echo off  
set /p name="What is your name? "  
echo Hello %name%  
  
portfolio > week 11 > hello.bat  
1  @echo off  
2  set /p name="What is your name? "  
3  echo Hello %name%  
4  
  
Ln 4, Col 1  Spaces: 4  UTF-8  CRLF  Batch  ☺  🔔
```

2. Run this script using the cmd command line and ensure that the output is correct.

Example Output:

```
Administrator: C:\Windows\system32\cmd.exe  
C:\CSI6203\CSI6203>cd "portfolio\week 11"  
C:\CSI6203\CSI6203\portfolio\week 11>hello.bat  
What is your name? Joe  
Hello Joe  
C:\CSI6203\CSI6203\portfolio\week 11>
```

3. Now that the environment is set up and working, create a new batch script named “FolderMaker.bat” which prompts the user to type the name of a folder to create.
4. The new folder should then be created inside the “scratch” folder in your portfolio.

Example Output:

```
C:\CSI6203\CSI6203\portfolio\week 11>FolderMaker.bat
type the name of a folder to create: new

C:\CSI6203\CSI6203\portfolio\week 11>dir ..\..\scratch
Volume in drive C is Windows 10
Volume Serial Number is B009-E7A9

Directory of C:\CSI6203\CSI6203\scratch

08/24/2019  03:55 AM      <DIR>        .
08/24/2019  03:55 AM      <DIR>        ..
08/24/2019  03:54 AM      <DIR>        new
              0 File(s)       0 bytes
              3 Dir(s)   25,886,224,384 bytes free

C:\CSI6203\CSI6203\portfolio\week 11>
```

Task 5: Writing a simple PowerShell script

1. Before we can write PowerShell scripts, we need to set the Execution Policy to allow them to be run.
2. Open an administrator PowerShell command prompt through the start menu and type the following command:

```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Windows\system32> Set-ExecutionPolicy Unrestricted
```

3. This will prompt you to accept the new execution policy, Type 'Y' for "yes".

```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Windows\system32> Set-ExecutionPolicy Unrestricted

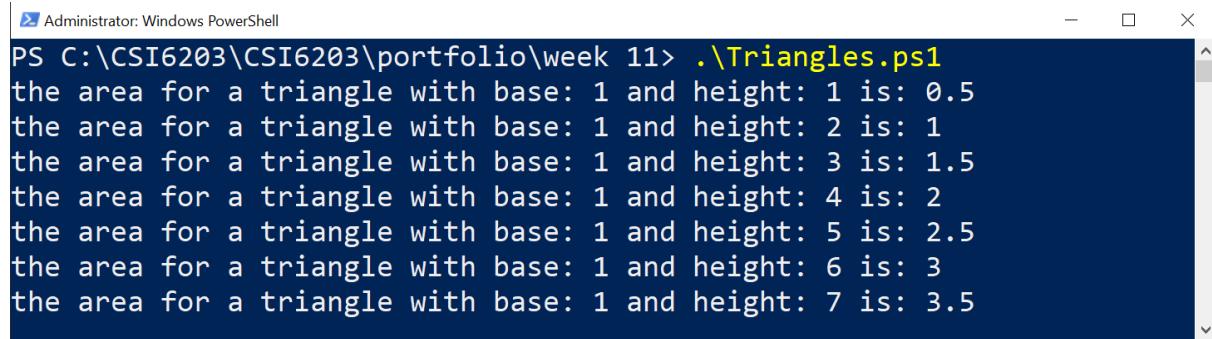
Execution Policy Change
The execution policy helps protect you from scripts that you do not
trust. Changing the execution policy might expose you to the security
risks described in the about_Execution_Policies help topic at
https://go.microsoft.com/fwlink/?LinkID=135170. Do you want to change
the execution policy?
[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help
(default is "N"):
```

4. Now create a new PowerShell script named “Triangles.ps1” that replicates the behaviour of “Triangles.sh” from week 6.

Note : You'll need to write in “PowerShell” scripting instead of “bash” or “Batch” scripting.

Hint: use for loops in PowerShell to achieve the same kind of output!

Example output:



```
Administrator: Windows PowerShell
PS C:\CSI6203\CSI6203\portfolio\week 11> .\Triangles.ps1
the area for a triangle with base: 1 and height: 1 is: 0.5
the area for a triangle with base: 1 and height: 2 is: 1
the area for a triangle with base: 1 and height: 3 is: 1.5
the area for a triangle with base: 1 and height: 4 is: 2
the area for a triangle with base: 1 and height: 5 is: 2.5
the area for a triangle with base: 1 and height: 6 is: 3
the area for a triangle with base: 1 and height: 7 is: 3.5
```

Note that PowerShell can do non-integer division operations without needing any extra programs!

Task 6: git and github

1. Now that we have the next scripts completed for our portfolio, you can commit your local changes to the git repository.
2. The commands for this are the same in Windows as in Linux.
3. Make sure to add all of the new work to source control.
 - [hello.bat](#)
 - [Foldermaker.bat](#)
 - [Triangles.ps1](#)