Name: Nicholas Defranco

Student #: 106732183

\*\*\*I did some work on a lab machine and some on my personal computer which is why drive letters are different, I worked on a USB\*\*\*

1. What was the response to the **get-history** command? (The answer isn’t nothing if you opened PowerShell using File Explorer as described above.)   
   🡻

Id CommandLine

-- -----------

1 Set-Location -literalPath 'E:\CPR101\_Activities\Week8'

2 get-history

*Note:* to find a previously run command, use the up arrow key to recall commands from the history stack to the command line. To see all previously run commands, use get-history at any time.

1. Enter the command: **get-help**You will see this on a lab computer or possibly on your own laptop:  
   **Do you want to run Update-Help?  *yes, you do***   
   The Update-Help cmdlet downloads the most current Help files for Windows PowerShell modules, and installs them on your computer. For more information about the Update-Help cmdlet, see http://go.microsoft.com/fwlink/?LinkId=210614.  
   [Y] Yes [N] No [S] Suspend [?] Help (default is "Y"): \_ **🡨 Enter Y=Yes**This will take about 30 seconds.  
   What is the **SHORT DESCRIPTION** that was output? (scroll up to see all the help text)  
   🡻

Displays help about Windows PowerShell cmdlets and concepts.

1. PowerShell can get information about various system processes or services.   
   Try > **Get-Process**  
   Now try > **Get-Process | sort-object CPU -descending | select -first 10**and copy/paste the output here. You can probably guess what the command will do.  
   It shows the top 10 processes using the most CPU seconds.  
   🡻

Handles NPM(K) PM(K) WS(K) CPU(s) Id SI ProcessName

------- ------ ----- ----- ------ -- -- -----------

357 22 47588 60528 1,368.36 7728 0 TMBMSRV

1301 66 77128 16868 706.80 4704 0 Ntrtscan

4113 0 196 1916 414.13 4 0 System

391 14 8488 18036 192.67 2708 0 svchost

55 5 3120 3628 168.42 6320 0 cmd

334 10 3180 9276 149.31 1356 0 svchost

1124 36 52776 8992 141.28 3972 0 TmCCSF

1616 35 9028 22540 70.25 844 0 lsass

1185 20 12360 19348 58.89 1304 0 svchost

832 42 98228 115960 46.86 10320 1 powershell

FYI, the equivalent command on a Linux/Unix system, is very cryptic (and will not run in PS):  
> ps aux | sort -nrk 3,3 | head -n 10

1. Enter and run each of these three commands, one at a time:   
   **> dir   
   > ls   
   > get-childitem**If there is no output, it means there is nothing to list.   
   A good interface would say "This folder is empty." like File Explorer does.  
   If there is no output, **navigate to the Documents folder *or any folder that is not empty* and repeat.**  
   Do you recognize any of these commands? Did they all do the same thing? What did they do?   
   🡻

I do recognize these commands

ls is the command in \*nix to list files in the directory specified (current directory is default).

dir is the command in the CMD in windows that also lists the files in the directory specified (current is default).

* These two commands are aliases.

The output is the same for each, it provides the programmer with file type (either it is directory, archive, read-only, hidden, system), current file permissions, timestamp, size of file, and name in order as mentioned.

1. Copy the first 10 lines of output from the last of those commands and paste here.  
   🡻

Mode LastWriteTime Length Name

---- ------------- ------ ----

d----- 2018-10-19 9:34 AM APC

d----- 2018-10-03 8:24 AM IPC

d----- 2018-10-03 8:21 AM CPR101\_Activities

d----- 2018-06-17 8:37 PM Sumo

d----- 2018-09-12 10:08 AM MyFirstApplication

-a---- 2018-10-28 7:33 PM 14108 EAC149\_Assignment.docx

-a---- 2018-10-03 1:08 PM 66 README.txt

-a---- 2018-10-31 6:31 PM 4 test.txt

1. Enter the command: **get-childitem -recurse**What was different about the output this time?  
   🡻

It looked inside the directories if it is a directory and listed its contents. If those are directories it will look inside that and so on (lists recursively).

1. Enter the command: **get-childitem -recurse > desktopItems.txt**What was different about the output this time? (If not sure what happened, see the next task.)  
   🡻 (the answer isn’t “nothing”)

The standard output of the command was redirected to a new text file (if it didn’t exist) called desktopItems.txt, the single greater than (inequality symbol) overwrites this file if it already existed with content.

1. Search for the desktopItems.txt file using the TAB key.  
   Type: **\*.txt***[press TAB key]* or type: **des***[press TAB key]**Note: do not type a space following the* ***\*.txt*** *or* ***des*** *characters, just press the TAB key.*Depending on the number of files in the Desktop folder matching the **\*.txt** wildcard or files beginning with **des**, you may have press the TAB key multiple times. Once you see **.\desktopItems.txt** *press the Enter key*. What happened?  
   🡻

It opened notepad with the output of the **Get-Children –Recurse** command that was redirected before from a previous command.

More generally it opens the default application that is used to read and edit text files.

1. Make a copy of the desktopItems.txt file. To find the command to copy a file:  
   Remember the **verb-noun** or **action-object** format, hyphenated without a space.  
   > **get-help copy\*** *will find PowerShell commands starting with “copy”* One of them is used to copy a file.> **get-help** PowerShellC*ommandName*  will show a command’s parameters  
   When entering parameters for the copy command, typing **des***[TAB,TAB,TAB,…]* will insert the name of a file beginning with "des". This avoids typing mistakes when identifying your desktopItems.txt filename.  
   What is the PowerShell command, including parameters, to copy that file?   
   🡻

The command is Copy-Item.

The basic syntax is:

Copy-Item <target> <destination>

🡻 🡻

File (of any type where the contents to

like directory) to should be copied

be copied and name of file

\*\*\*parts in the <> are meant to be replaced\*\*\*

Example:

Copy-Item .\desktopItems.txt .\files.txt

* This copies the contents of text file desktopItems.txt(in the current directory) to another file called files.txt(overwrites it if it already exists with test inside).

1. For the delete command, what is the PowerShell name and what are all its aliases?   
   > **get-alias del** (or **rm**) will show the PowerShell name for the DOS or \*nix command. To find all aliases for a PowerShell command name,   
   > **get-alias -Definition**  PowerShellC*ommandName*  So, what is the PowerShell command to delete/remove a file and what are *all* its aliases?  
   🡻

The command to delete/remove a file is: Remove-Item

Aliases

* ri
* rm
* rmdir
* del
* erase
* rd

1. Delete the **desktopItems.txt** file.   
   What was the PowerShell command, including parameters, you ran to delete the file?   
   🡻

The PowerShell command to delete a file is: Remove-Item

The basic syntax is:

Remove-Item <target>

🡻

File (of any type

like directory) to be

Removed

For example:

Remove-Item .\desktopItems.txt

This removes the desktopItems.txt file, this command also removes directories given that one of the aliases is rmdir.

1. Finally, run the **get-history** command again. Copy the output and paste here.  
   🡻

Id CommandLine

-- -----------

1 Set-Location -literalPath 'D:\CPR101\_Activities\Week8'

2 ls

3 Get-Help copy

4 get-alias rm

5 get-help rm

6 get-help del

7 get-alias del

8 get-help cd

9 cd ..

10 ls

11 cd ..

12 ls

13 Remove-Item .\desktopItems.txt

14 ls

15 get-history

16 Get-Help

17 Get-Process

18 Get-Process | sort-object CPU -descending | select -first 10

19 dir

20 ls

21 Get-ChildItem

22 get-childitem -recurse

23 Get-ChildItem -recurse > desktopItems.txt

24 .\desktopItems.txt

25 Get-Help copy

26 get-alias del

27 get-alias rm

28 get-alias -Definition remove\*

29 get-alias del

30 Remove-Item .\desktopItems.txt

**Part 2: Time Management**

What **four** (or more) things will you do to be more successful with your time?   
(40 points)The lecture notes and above resources may provide some insights. There are no right answers here except the answers that are right for you. Four simple one line bullets are worth 10 points. Explanations of why you will take action on those points today and how you will keep taking those actions over time is worth the other 30 points.

I used these links that you provided to help decide my four things to do to be more successful.

<https://www.youtube.com/watch?v=tf9ZhU7zF8s>.

[www.timewellspent.io](http://www.timewellspent.io/)

[Take Control](http://www.timewellspent.io/take-control/)

[Demand Better Design](http://www.timewellspent.io/designers/)

<https://www.youtube.com/watch?v=N4YVLkuRBe8>

As well as your version control PowerPoint

**Please delete the Part 2 instructions above before submitting.**

🡺 At the end of every school day, I will reflect on what was everything that was assigned as well as assignments that are incomplete for previous days, and determine based on the instructions, how long it would take to complete to assignment. I will also look at the due date given for it to determine what I should be working on at what time. I can also plan where I can do this work. I will do this because this approach gives me a schedule to know when exactly I will be working and waste very little time such as allocating study/work time between classes as mentioned in ThePenguinProf video (PenguinProf 5:35) as I will also know where to be to maximize efficiency in my work such as a quiet place such as the library. This will be a valuable skill to gain which can used in the workforce.

🡺 When I am working, I will remove distractions as much as possible such as a cell phone. I can mute my notifications on my phone so there are no interruptions. If I am using my computer, I will avoid using any distracting applications by hiding them such as not having any shortcuts to any games on my desktop this way I am not tempted to play games at times I allocated for study time as mentioned on the last point. I can also mute notifications on my laptop (windows 10 notifications) to avoid opening my advertisements as well as YouTube videos (chrome notifications) as mentioned in humantech.com. I will do this because device notifications, as mentioned in Time Well Spent video (Stossel 0:30), are specifically made to continuously draw your attention to your phone. By hiding this feature, it will be hard for the app to draw your attention. This makes it very easy to stop mindlessly using applications now and when I continue my studies in the future.

🡺 To continue from the first point, along with creating a schedule, I will also create it so that is flexible. By achieving flexibility, if for example a friend wants to get together at a time you originally planned to study, instead of saying no right away I can rearrange my schedule to give me time to get together with that friend (if they do not do it often and interrupt the best times for me study). This is important because it is a proven fact that the more freedom someone has with their life, the happier they are as mentioned in the ThePenguinProf video (PenguinProf – entire idea). To make my schedule more flexible, I will leave a few hours on my schedule left out without any tasks this could end up in free time if nothing comes up.

The next point your version control PowerPoint into consideration.

🡺 This relates with programming. To be more successful with making a project I can get used to using git or GitHub to help with multiple versions of the same file or files each with a hopefully meaningful description where the first version only contains comments describing what goes where and the desired output. If anything has went wrong with a newer version, I do not have to worry as I can easily revert to an old version that was committed to the project (hopefully this will not happen, and everything was tested). Every new small feature that is added will be tested then committed (along with comments in the code that help to plan out the code). This is important because by doing small changes and testing often, this ultimately results in less debugging time searching for sematic errors in a program (Software Development slide 17). Git will take some practice but once I get the hang of it, I will be doing every project of on it improving my time needed to make a project as will not need to manually create a new version for every major/minor/micro feature enhancement.

Extra

\*\*\* I plan my programming projects very vaguely and I tend to forget some things I need to get the desired output. This means I forget they mention things in the comments that should be placed before any code. I should take more time into thinking about what would need to get the desired output from my project especially when my projects get bigger. \*\*\*