**Part I: Questions**

What is a Graphical User Interface (GUI) and how does that differ from the Command Line Interface (CLI)?

The**main difference**between GUI and CLI is that**the Graphical User Interface (GUI) allows the user to interact with the system using graphical elements such as windows, icons, menus while the Command Line Interface (CLI) allows the user to interact with the system using commands.**

**What does the shell do?**

**A shell manages the user–system interaction by prompting users for input, interpreting their input, and then handling an output from the underlying operating system; in other words, the shell is a program that takes commands from the keyboard and gives them to the operating system to perform.**

**What is the command used for listing things in a directory?**

**ls**

**What is the command used for changing directories?**

**cd**

**What command would you use to get your current working directory?**

**pwd**

**How do you get the manual for these commands?**

**By using man**

**What does the shell prompt look like?**

**Looks like a command line interface, similar to MSDOS (very old).**

**How would you list things in a directory in chronological order?**

**ls –t (sort by time modified) or ls –U (sort by time of file creation) or ls –lt (chronological)**

**Name two ways to get to the ‘home’ or ‘root’ directory.**

**cd and ~**

**What is the difference between an absolute path and a relative path?**

**An absolute path always contains the root element and the complete directory list required to locate the file, example: /Users/santiagovillamarin/Desktop/data-shell/creatures. A relative describes where a folder or file is in relation to another folder, example: ~/data-shell/creatures.**

**What are the two relative path directories we talked about and what do they mean?**

**. = my current directory**

**.. = up a directory**

**Why shouldn’t you put spaces in filenames?**

**Because the space is used as a delimiter, separating the name from the arguments, so many errors might come from that confusion when reading files.**

**Name a way to have multiple readable words in a filename without spaces?**

**FileNameNumber.txt**

**What is the program nano, what does it do? Do you use nano? If not what do you use?**

**Nano creates a new file to be edited, I use it to create executable python files.**

**\* is a form of a wildcard. What does it mean?**

**It means that it will read any file, for instance if I add \*.txt, it will list or read all the text files in a directory.**

**# \* just matches any character**

**The following code would match what? ls \*.txt**

**Will make a list of all the text files in the directory**

**Why do we need to be careful with the mv command?**

**Because you are able to move any file or rename it and if you are not paying attention a whole directory can be moved and lost.**

**Why do we need to be careful with the rm command?**

**Because you can remove complete directories and without a chance to recovery.**

**What is the difference between > and >>?**

**> Send output of a command to a file, overwriting existing files**

**>> Send output of a command to a file, appending to existing files**

**What does head do?**

**Show the first lines of a file or command**

**What is the purpose of | (pipe)?**

**Pipe output of one command to the input of another command**

**Interpret the following command:**

**cat huge\_file.fasta | uniq | head -n 5 >proteins.fasta**

**It will filter all the information from huge\_file.fasta, filtering out adjacent matching lines in the file using uniq, and then counting only the first five lines from the file. All the command output will be redirected to a new file called proteins.fasta**

**What is a loop and when would you use it?**

**Loops are a programming construct, which allow us to repeat a command or set of commands for each item in a list. As such, they are key to productivity improvements through automation.**

**We start by adding a “for” then a name (any name) “file” in then add the filename to be used “doc.txt”, then say “do” and add a description of what we want to do and at the end we say “done” or “print”**

**In a directory with the following files. What would the following loop do? apple.txt, banana.txt, orange.txt $ for filename in \*.txt; do > cat ${filename} >>fruits.txt > done**

**It will concatenate all the contents from the three .txt files into one called “fruits”.**

**What is a shell script?**

**A shell script is just a text file that contains a list of shell commands that you want to run in sequence.**

**What are the benefits of writing and running a script over typing the code in?**

**I did not get this question, what I found was that often, writing a shell script is much quicker than writing the equivalent code in other programming languages.**

**#creating a script means you wont have to type the code in yourself every time – you can just run your script – it is a way to save code so you don’t have to retype it.**

**When looking at a script what does # mean? Why would you use one?**

**# Is used to write messages that would not run as a command, I would use it to document what the script means and does.**

**What does grep stand for and what does it do?**

**Global Regular Expression Print**

**What does grep do?**

**Searches for phrase in a list of files or pipe and show matching lines and searches for a string of characters in a specified file. The text search pattern is called a regular expression. When it finds a match, it prints the line with the result. The grep command is handy when searching through large log files.**

**What does find do?**

**The find command in UNIX is a command line utility for walking a file hierarchy. It can be used to find files and directories and perform subsequent operations on them. It supports searching by file, folder, name, creation date, modification date, owner and permissions.**

## Part II - Writing Code

**Open the shell and change directories to the desktop and list the contents, paste your code here.**

**Santiagos-MacBook-Air:planets santiagovillamarin$ cd**

**Santiagos-MacBook-Air:~ santiagovillamarin$ cd Desktop**

**Santiagos-MacBook-Air:Desktop santiagovillamarin$ ls**

**1\_deQueiroz\_2007\_Species Concepts.pdf**

**COS Travel Worksheet.xlsx**

**CV Betzabet Obando.pdf**

**Cedula.docx**

**Comps/**

**DTM/**

**Draw the file structure from the root directory into one of the folders on the desktop. You can submit a pdf with a drawing. Select three locations and type the commands for changing directories to those locations use a combination of absolute and relative paths.**

**/Users/santiagovillamarin/Desktop/Hwork1/Science.txt**

**/Users/santiagovillamarin/Desktop/planets/Mars.txt**

**Desktop/Hwork1/Data.txt**

****

**On your computer create a directory on your desktop. Create two files without opening them that have file extensions .txt. Then use wildcards to list the files in that directory. Paste you code here.**

**Santiagos-MacBook-Air:Desktop santiagovillamarin$ mkdir Hwork1**

**Santiagos-MacBook-Air:Desktop santiagovillamarin$ cd Hwork1**

**Santiagos-MacBook-Air:Hwork1 santiagovillamarin$ touch Data.txt**

**Santiagos-MacBook-Air:Hwork1 santiagovillamarin$ touch Science.txt**

**Santiagos-MacBook-Air:Hwork1 santiagovillamarin$ ls -1 \*.txt**

**Data.txt**

**Science.txt**

**Download the file Hutia\_DNA.fasta from the Github repo. Make a new directory on the desktop. Move the file into that directory. List the contents of the directory. How many lines are in this file? Fasta files are a type of data file that holds DNA sequences. They are all formatted the same. The first line starts with a > and a name followed by a line break, and then DNA sequence followed by a line break. The next line starts with an > and another name followed by a line break and the DNA sequence. With that in mind, how could you use linux commands to determine how many sequences are in this file? How many sequences are in this file?**

**Santiagos-MacBook-Air:Hwork1 santiagovillamarin$ cd ~/Desktop**

**Santiagos-MacBook-Air:Desktop santiagovillamarin$ mkdir Hutia**

**Santiagos-MacBook-Air:Desktop santiagovillamarin$ cd ~/Downloads**

**Santiagos-MacBook-Air:Downloads santiagovillamarin$ mv Hutia\_DNA.fasta ~/Desktop/Hutia**

**Santiagos-MacBook-Air:Downloads santiagovillamarin$ cd ~/Hutia**

**Santiagos-MacBook-Air:Hutia santiagovillamarin$ ls**

**Hutia\_DNA.fasta**

**Santiagos-MacBook-Air:Hutia santiagovillamarin$ wc -l Hutia\_DNA.fasta**

**2382004 Hutia\_DNA.fasta #Total number of lines**

**Santiagos-MacBook-Air:Hutia santiagovillamarin$ grep ">" -c Hutia\_DNA.fasta**

**1191002 #Total number of sequences**

**Build a single line of code that would take the first 100 sequences of this file and put them in a new file called ‘MyOutputFile.fasta’. Use a combination of wc, sort and uniq in a single line of code to tell me how many unique lines are in this file. Paste your commands and the answer here.**

**Santiagos-MacBook-Air:Hutia santiagovillamarin$ grep ">" Hutia\_DNA.fasta | uniq | head -n100 | sort -n > MyOutputFile.fasta**

**Write code that would create a loop to copy all the files in one directory ending in .txt to another directory. Paste the code here.**

**For files in \*.txt**

**do**

**cp $files ~/Hutia > ~/Data**

**done**

**#I don’t think you need the > but otherwise great!**

**Write a bash script with in-line documentation (hint #) to show how to run a made up python program (script) called 'find\_taxa.py' on set of files ending in ‘.fasta’. Show in the script how to run the program on each of those files. What is the name of your script? Type the script here.**

**touch find\_taxa.py**

**chmod u+x find\_taxa.py**

**nano find\_taxa.py #opens text file to create the script**

**#DO NOT know exactly what we want to do for each .fasta file, SORRY!!!**

**python3 find\_taxa.py #will run the script**

**for file in \*.fasta**

**do**

**python find\_taxa.py $file**

**done**

**From the Hutia\_DNA.fasta file tell me how on how many lines do we find the pattern ‘GAGA’. What was the code used to find this?**

**Santiagos-MacBook-Air:Hutia santiagovillamarin$ grep GAGA Hutia\_DNA.fasta | wc -l**

**304806**