## 

## 

**Experiment 1:**

**Using Software Tools and Code Versioning System**

CPE106L (Software Design Laboratory)

**Manzanero, Jan Adrian C.**

Group No.: **6**

Section: **B1**

## **PreLab**



|  |
| --- |
| **Readings, Insights, and Reflection** |

GitHub is like a central online storage for computer code. It keeps track of different versions of your code as you make changes so you can see its history and go back to an earlier version if needed. Multiple programmers can work on the same project on GitHub simultaneously, like a shared workspace. You can share your code publicly for others to learn from or keep it private. GitHub is a popular platform for open-source projects where anyone can contribute, making it a great place for programmers to work together and build cool things.

The lab session focused on equipping students with the essentials of using GitHub for software development. Students set up their accounts and learned practical skills like creating repositories (cloning) to store their code. They explored the concept of branching within a repository, allowing them to experiment with changes without affecting the main project. Additionally, the session covered how to collaborate by adding others to work on the same repository. Finally, students were introduced to using the command line interface (CLI) to push their code changes (commits) to the GitHub repository. This hands-on experience provided a solid foundation for the students to delve deeper into the functionalities of GitHub for future software projects.

Overall, GitHub can be effectively utilized in schools and workplaces for its version control capabilities and potential for collaboration. Using GitHub allows classmates to work on the same project simultaneously, see each other's contributions, and streamline the development process. In essence, the lab session equipped the students with foundational skills in GitHub that promote organized and collaborative software development.

## **InLab**



**Objectives**

* Check the following GitHub learning lab courses:

<https://github.com/skills/introduction-to-github>

* Follow the steps to complete the course as shown in the figures below.
* Follow the Lab Report Template InLab Guide on what to include (i.e. discussion, screenshots, etc.).

**Tools Used**

* GitHub
* Oracle Virtual Machine VirtualBox Manager
* Visual Studio Code

**Procedure**

**Part 1: Introduction to GitHub**

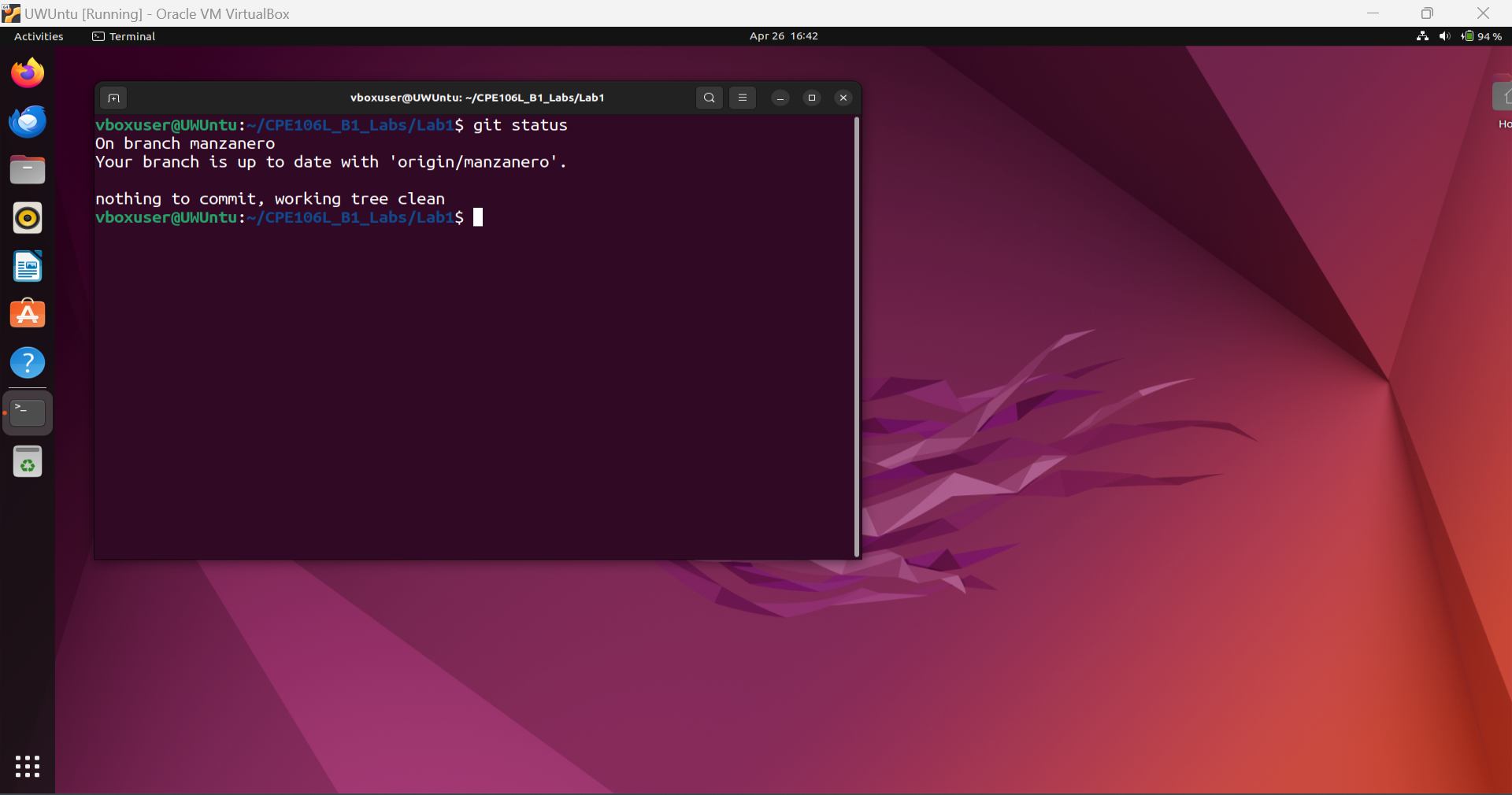
To begin using GitHub, the student must first create a GitHub account using their Mapúa email address. Upon registering their account, the students were asked to provide their profile information and a profile picture to help the instructor identify the student.

A screenshot of a computer

Description automatically generated

*Figure 1.1. The student’s GitHub profile page.*

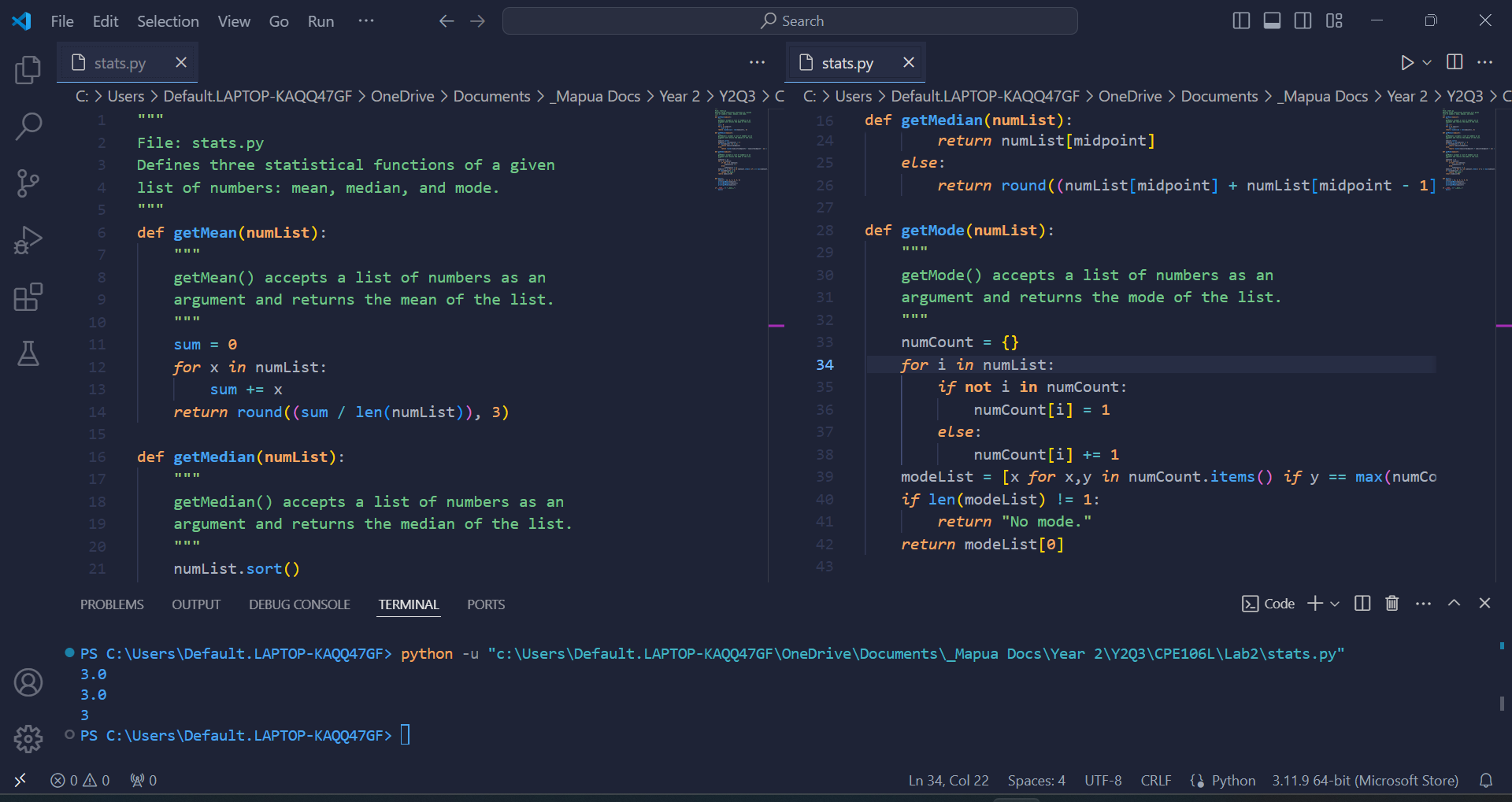
The GitHub account should be connected to the student’s VM to set up a local repository on the student’s side and utilize the Git commands in the Linux terminal.



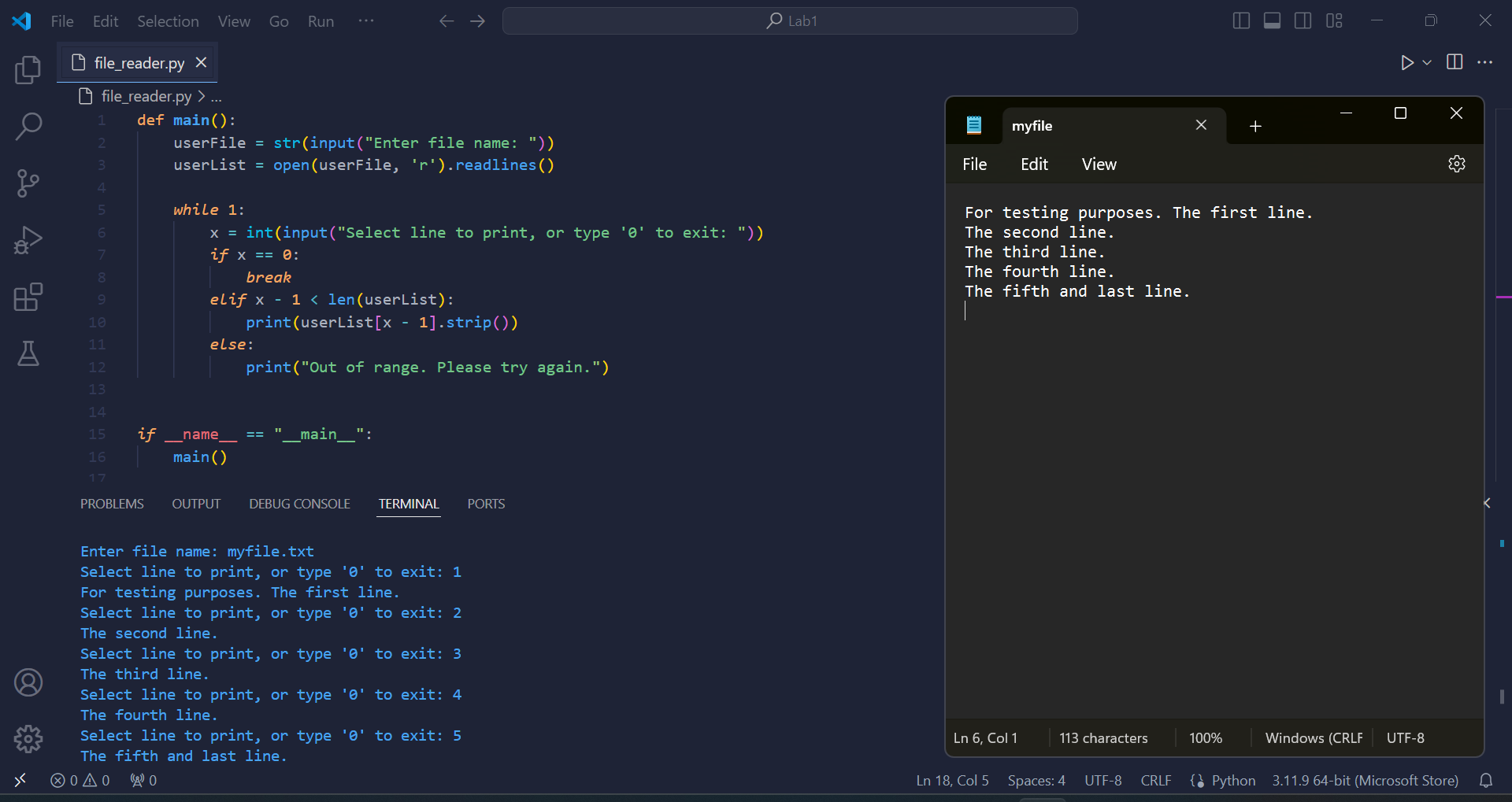
*Figure 1.2. Connecting the GitHub account to the Linux terminal.*

**Part 2: Introduction to Python**

Snippets of the solutions for the Lab1 Programming Problems are tested in Visual Studio Code and displayed below.



*Figure 2.1. Sample output for* stats.py *with* numList = [1, 2, 3, 3, 4, 5]*.*



*Figure 2.2. Sample output for* file\_reader.py *with test file* myfile.txt*.*

## **PostLab**



GitHub Repository Branch:

**Programming Problem 1:**

def mean(numList):

return sum(numList)/len(numList)

def median(numList):

length = len(numList)

i = length // 2

# for odd list count = one median

if length % 2 == 1:

return(numList[i])

# for even list count = average of medians

return(sum(numList[i - 1: i + 1])/2)

def mode(numList):

numCount = {}

for i in numList:

if not i in numCount:

numCount[i] = 1

else:

numCount[i] += 1

modeList = [x for x,y in numCount.items() if y == max(numCount.values())]

if len(modeList) != 1:

return "No mode."

return modelist[0]

**Programming Problem 2:**

def main():

userFile = str(input("Enter file name: "))

userList = open(userFile, 'r').readlines()

while 1:

x = int(input("Select line to print, or type '0' to exit: "))

if x == 0:

break

elif x - 1 < len(userList):

print(userList[x - 1].strip())

else:

print("Out of range. Please try again.")