



MAPUA UNIVERSITY

SCHOOL OF ELECTRICAL, ELECTRONICS, AND COMPUTER ENGINEERING

Experiment 1:

Using Software Tools and Code Versioning System

CPE106L (Software Design Laboratory)

Member 1: Aljehro R. Abante

Member 2: Cerdan Karl T. Alarilla

Member 3: Daryl Jake A. Fernandez

Group No.: **3**
Section: **E01**



PreLab

Readings, Insights, and Reflection

Introduction to Git and Professional Git

<http://guides.beanstalkapp.com/9781119285007>

- The documents helped us discover that git offers both command-line and client-based interactions. We've realized that mastering the command lines enhances our understanding of how Git works. The reading covers the essential commands for managing local repositories like git init to start, git add to stage our changes, and git commit to record modifications. Overall, this guide is a great resource that made our Git experience easier.

Fundamentals of Python: Data Structures , 2nd Edition 9780357122754

- In our exploration of Fundamentals of Python: Data Structures, we learned of the core of Python programming and applying it more in a practical way. We learned to write basic programs, handle input and output, and apply arithmetic and logical operations. The reading taught us more about data structures (lists, tuples, and dictionaries). On the other hand, the control statements in Python lets us use our logic to code. We also learned about IDLE which is used to compile and run codes interactively. The naming conventions also taught us how to differentiate and create descriptive names for variables and functions.

Git, Anaconda, and Visual Studio Code

- We've learned some essential things about Git, Anaconda, and Visual Studio Code. Git enabled us to improve teamwork and maintain our code by gaining knowledge about tracking changes, co-authoring with others, and managing versions of the same code. Anaconda provided us with a powerful environment for handling Python packages and dependencies, particularly relevant to Python programming and data science. This enabled us to create isolated environments that were tailored to particular projects or task types. Visual Studio Code was the preferred choice for writing, debugging, and testing code in a variety of programming languages due to its user-friendly interface and extensive capabilities. These tools allowed us to work more cohesively, handle large-scale assignments, and improve our coding skills.

InLab

- **Objectives**

1. Introduction to GitHub
2. Introduction to Python

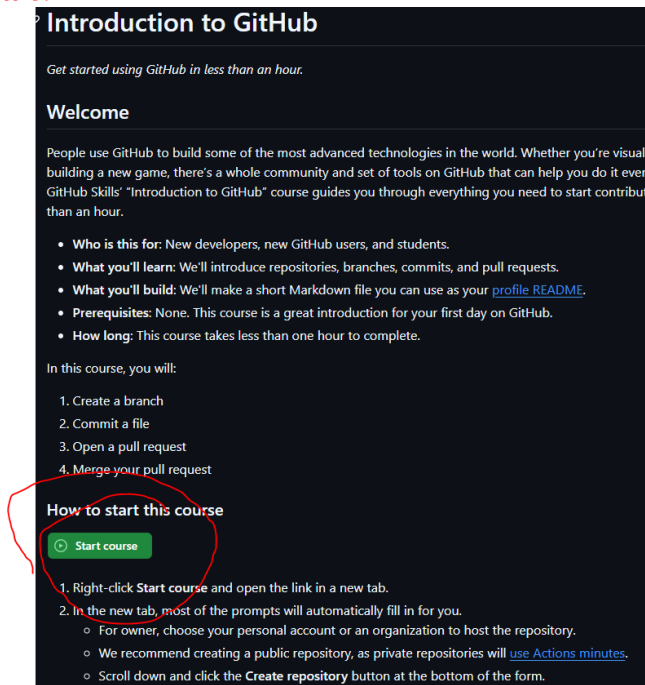
- **Tools Used**

- Anaconda/Python/VS Code
- Git Terminal/ Ubuntu/wsl

- **Procedure**

GitHub

1. Go to Git tutorial link. After getting tutorial link, right click Start Course button and open in new tab:



2. Click "Create Repository" button:

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk ().*

Repository template

skills/introduction-to-github ▾

Start your repository with a template repository's contents.

☐ **Include all branches**
Copy all branches from skills/introduction-to-github and not just the default branch.

Owner * **Repository name ***

✔ skills-introduction-to-github is available.

Great repository names are short and memorable. Need inspiration? How about **cautious-happiness** ?

Description (optional)

☒ **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

① You are creating a public repository in your personal account.

Create repository

3. Create a branch and name it to new branch:

This branch is up to date with **main**.

Contribute ▾

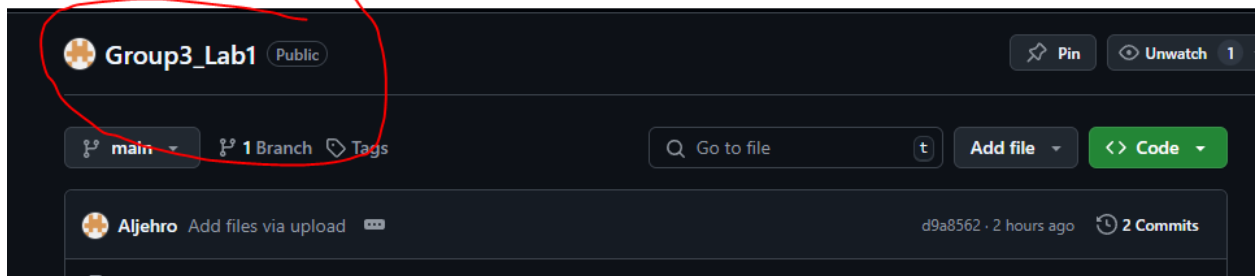
File	Commit Message	Commit Hash	Time Ago
github-actions[bot]	Update to 1 in STEP and README.md	ac661b8	1 minute ago
.github	Update to 1 in STEP and README.md		1 minute ago
images	Initial commit		1 minute ago
.gitignore	Initial commit		1 minute ago
LICENSE	Initial commit		1 minute ago
README.md	Update to 1 in STEP and README.md		1 minute ago

README License

Introduction to GitHub

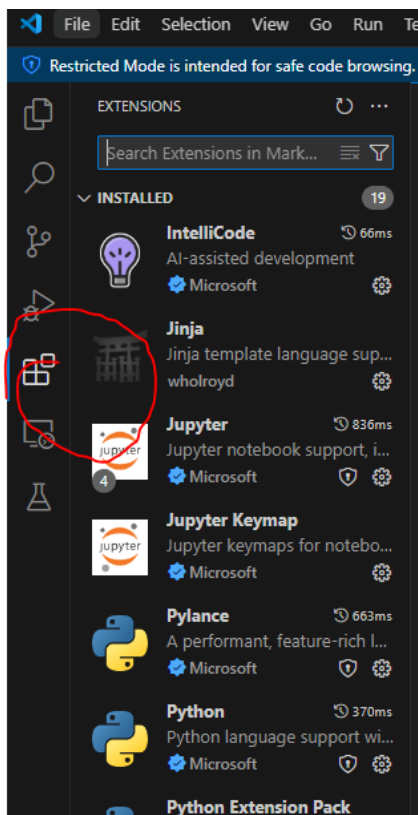
Get started using GitHub in less than an hour.

4. With the tutorial finished, create a repository for Lab1:



Introduction to Python

1. Install VS Code
2. In the left side, click the extensions and install Python libraries:



3. In WSL Ubuntu terminal, install Anaconda. Then proceed with the rest of Lab1.

```
(base) aljehro@DESKTOP-FUQQ73K:~/LocalRepo$ cd cpe106l-4/  
(base) aljehro@DESKTOP-FUQQ73K:~/LocalRepo/cpe106l-4$ python3  
Python 3.11.3 (main, Apr 19 2023, 23:54:32) [GCC 11.2.0] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>> print("hello world")  
hello world  
>>>
```

PostLab

- **Programming Problem 1:** Statisticians would like to have a set of functions to compute the median and mode of a list of numbers. The median is the number that would appear at the midpoint of a list if it were sorted. The mode is the number that appears most frequently in the list. Define these functions in a module named stats.py. Also include a function named mean, which computes the average of a set of numbers. Each function expects a list of numbers as an argument and returns a single number.

```
• # CPE106L-4/E01
• # Lab1 || Postlab #1
• # Group #3 Members: Abante, Alarilla, Fernandez
•
• #stats.py
• def mean(numbers):
•     return sum(numbers) / len(numbers)
•
• def median(numbers):
•     sort_num = sorted(numbers)
•     n = len(sort_num)
•     mid = n // 2
•     if n % 2 == 0:
•         return (sort_num[mid - 1] + sort_num[mid]) / 2
•     return sort_num[mid]
•
• def mode(numbers):
•     frequency = {}
•     for number in numbers:
•         frequency[number] = frequency.get(number, 0) + 1
•     max_freq = max(frequency.values())
•     modes = [k for k, v in frequency.items() if v ==
max_freq]
•     return modes
•
• def main():
•     while True:
•         user_input = input("Enter a list of numbers
separated by spaces. (Type 'q' to quit): ")
•         if user_input.lower() == 'q':
•             print("Terminating program.....")
•             break
•
```

```

•         try:
•             data = list(map(float, user_input.split()))
•             print("Data:", data)
•             print("Mean:", mean(data))
•             print("Median:", median(data))
•             print("Mode:", mode(data))
•         except ValueError:
•             print("Invalid input.")
•
•     if __name__ == "__main__":
•         main()
•
•

```

Output:

```

aljihro@DESKTOP-FUQQ73K:~/LocalRepo/cpe106l-4/lab/Group3_Lab1$ conda activate
(base) aljihro@DESKTOP-FUQQ73K:~/LocalRepo/cpe106l-4/lab/Group3_Lab1$ python 3 stats.py
python: can't open file '/home/aljihro/LocalRepo/cpe106l-4/lab/Group3_Lab1/3': [Errno 2] No such file or directory
(base) aljihro@DESKTOP-FUQQ73K:~/LocalRepo/cpe106l-4/lab/Group3_Lab1$ python3 stats.py
Enter a list of numbers separated by spaces. (Type 'q' to quit): 1 2 3 4 5 6 21 21 2 3
Data: [1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 21.0, 21.0, 2.0, 3.0]
Mean: 6.8
Median: 3.5
Mode: [2.0, 3.0, 21.0]
Enter a list of numbers separated by spaces. (Type 'q' to quit): q
Terminating program.....
(base) aljihro@DESKTOP-FUQQ73K:~/LocalRepo/cpe106l-4/lab/Group3_Lab1$

```

- **Programming Problem 2:** Write a program that allows the user to navigate through the lines of text in a file. The program prompts the user for a filename and inputs the lines of text into a list. The program then enters a loop in which it prints the number of lines in the file and prompts the user for a line number. Actual line numbers range from 1 to the number of lines in the file. If the input is 0, the program quits. Otherwise, the program prints the line associated with that number.

```

# CPE106L-4/E01
# Lab1 || Postlab #2
# Group #3 Members: Abante, Alarilla, Fernandez

import os

def navigate_file():
    filename = input("Enter the filename: ")

    current_dir = os.path.dirname(os.path.abspath(__file__))
    file_path = os.path.join(current_dir, filename)

```



```

try:
    with open(file_path, 'r') as file:
        lines = file.readlines()

    num_lines = len(lines)
    print(f"The file contains {num_lines} lines.")

    while True:
        try:
            line_num = int(input("Enter a line number (0 to
quit): "))
        except ValueError:
            print("Please enter a valid number.")
            continue

        if line_num == 0:
            print("Exiting the program.")
            break
        elif 1 <= line_num <= num_lines:
            print(f"Line {line_num}: {lines[line_num -
1].strip()}")
        else:
            print(f"Please enter a line number between 1 and
{num_lines}.")

    except FileNotFoundError:
        print(f"The file '{filename}' was not found in the current
directory.")
    except IOError:
        print("An error occurred while reading the file.")

navigate_file()

```

Output:

```
(base) aljehro@DESKTOP-FUQQ73K:~/LocalRepo/cpe1061-4/lab/Group3_Lab/Lab1$ ls
PostLabSolution1 PostLabSolution2.py file.txt stats.JPG
(base) aljehro@DESKTOP-FUQQ73K:~/LocalRepo/cpe1061-4/lab/Group3_Lab/Lab1$ python3 PostLabSolution2.py
Enter the filename: file.txt
The file contains 6 lines.
Enter a line number (0 to quit): 1
Line 1: 1st Line: Group#3
Enter a line number (0 to quit): 2
Line 2: 2nd Line: Aljehro R.Abante
Enter a line number (0 to quit): 3
Line 3: 3rd Line: Cerdan Karl T. Alarilla
Enter a line number (0 to quit): 4
Line 4: 4th Line: Daryl Jake A. Fernandez
Enter a line number (0 to quit): 5
Line 5: 5th Line: Lab1_Group3.
Enter a line number (0 to quit): 6
Line 6: -----End of Line-----
Enter a line number (0 to quit): 0
Exiting the program.
(base) aljehro@DESKTOP-FUQQ73K:~/LocalRepo/cpe1061-4/lab/Group3_Lab/Lab1$
```