Week 9 Assignment

Name: Drishti Durgesh Telgu
Student ID: SM20240093
Unit : ICT_102
Professor: Dr. Duy Nguyen

Part 1: Understanding Different Data Manipulation Methods

Create a Python script named data_manipulation.py that demonstrates the following data manipulation methods:

Sorting: Sort a list of numbers.

Filtering: Filter out even numbers from a list.

Mapping: Square each number in a list.

Reducing: Compute the sum of a list of numbers.

```python
from functools import reduce


def sort_list(numbers):
   return sorted(numbers)


def filter_even(numbers):
   return list(filter(lambda x: x % 2 != 0, numbers))


def map_square(numbers):
   return list(map(lambda x: x ** 2, numbers))


def reduce_sum(numbers):
   return reduce(lambda x, y: x + y, numbers)
if __name__ == "__main__":
   numbers = [5, 3, 8, 6, 2, 7]

   print("Original list:", numbers)
   print("Sorted list:", sort_list(numbers))
   print("Filtered odd numbers:", filter_even(numbers))
```

```
    print("Squared numbers:", map_square(numbers))
    print("Sum of numbers:", reduce_sum(numbers))
```

Output:

```
C:\Users\61411\PycharmProjects\pythonProject\.venv\Scripts\python.exe
C:\Users\61411\PycharmProjects\pythonProject\.venv\Data_manipulation.p
y
Original list: [5, 3, 8, 6, 2, 7]
Sorted list: [2, 3, 5, 6, 7, 8]
Filtered odd numbers: [5, 3, 7]
Squared numbers: [25, 9, 64, 36, 4, 49]
Sum of numbers: 31

Process finished with exit code 0
```

Part 2: Understanding List Manipulation

Create a Python script named list_manipulation.py that demonstrates various list manipulation techniques, such as appending, inserting, removing, and slicing.

```
def list_manipulation_demo():
    my_list = [1, 2, 3, 4, 5]
    print("Original list:", my_list)
    my_list.append(6)
    print("After appending 6:", my_list)
    my_list.insert(2, 99)
    print("After inserting 99 at index 2:", my_list)
    my_list.remove(99)
    print("After removing 99:", my_list)
    sliced_list = my_list[1:4]
    print("Sliced list from index 1 to 3:", sliced_list)

if __name__ == "__main__":
    list_manipulation_demo()
```

Output:

```
C:\Users\61411\PycharmProjects\pythonProject\.venv\Scripts\python.exe
C:\Users\61411\PycharmProjects\pythonProject\.venv\list_manipulation.py
Original list: [1, 2, 3, 4, 5]
After appending 6: [1, 2, 3, 4, 5, 6]
After inserting 99 at index 2: [1, 2, 99, 3, 4, 5, 6]
After removing 99: [1, 2, 3, 4, 5, 6]
Sliced list from index 1 to 3: [2, 3, 4]

Process finished with exit code 0
```

Part 3: Writing a Palindrome and List Handling Program

Create a Python script named palindrome_list_handling.py that includes the following:

is_palindrome: A function that checks if a given string is a palindrome.

list_statistics: A function that computes the sum, average, and maximum of a list of numbers.

```python
def is_palindrome(s):
   s = s.lower().replace(" ", "")
   return s == s[::-1]


def list_statistics(numbers):
   if not numbers:
       return (0, 0, None)

   total = sum(numbers)
   average = total / len(numbers)
   maximum = max(numbers)
   return (total, average, maximum)
if __name__ == "__main__":
   string = "A man a plan a canal Panama"
   print(f"Is '{string}' a palindrome? {is_palindrome(string)}")
   numbers = [10, 20, 30, 40, 50]
   stats = list_statistics(numbers)
```

```
    print("List statistics:")
    print("Sum:", stats[0])
    print("Average:", stats[1])
    print("Maximum:", stats[2])
```

Output:

```
C:\Users\61411\PycharmProjects\pythonProject\.venv\Scripts\python.exe
C:\Users\61411\PycharmProjects\pythonProject\.venv\palindrome_list_han
dling.py
Is 'A man a plan a canal Panama' a palindrome? True
List statistics:
Sum: 150
Average: 30.0
Maximum: 50

Process finished with exit code 0
```

Part 4: Understanding Python Library and Package

Create a Python script named use_library.py that demonstrates the use of a Python library (such as math for mathematical functions) and creating a custom package.

Use the math library to compute the square root and power of a number.

Create a custom package with a module named mypackagethat contains a function greet.

For use_library.py :

```
import math
from my_package import greet

def demonstrate_library_usage():
    number = 16
    print(f"Square root of {number}:", math.sqrt(number))
```

```python
    print(f"{number} raised to the power of 2:", math.pow(number, 2))

if __name__ == "__main__":
    demonstrate_library_usage()
    greet.greet("User")
```

And for my_package/greet.py :

```python
import math
from my_package import greet

def demonstrate_library_usage():
    number = 16
    print(f"Square root of {number}:", math.sqrt(number))
    print(f"{number} raised to the power of 2:", math.pow(number, 2))

if __name__ == "__main__":
    demonstrate_library_usage()
    greet.greet("User")
```

Output:

```
C:\Users\61411\PycharmProjects\pythonProject\.venv\Scripts\python.exe
C:\Users\61411\PycharmProjects\pythonProject\.venv\use_library.py
Square root of 16: 4.0
16 raised to the power of 2: 256.0
Hello, User! Welcome to the Python package.

Process finished with exit code 0
```

Part 5: Working with Python GUI (Tkinter)

Create a Python script named simple_gui.py that creates a simple GUI application using Tkinter. The application should have:

A label that displays "Enter a number".

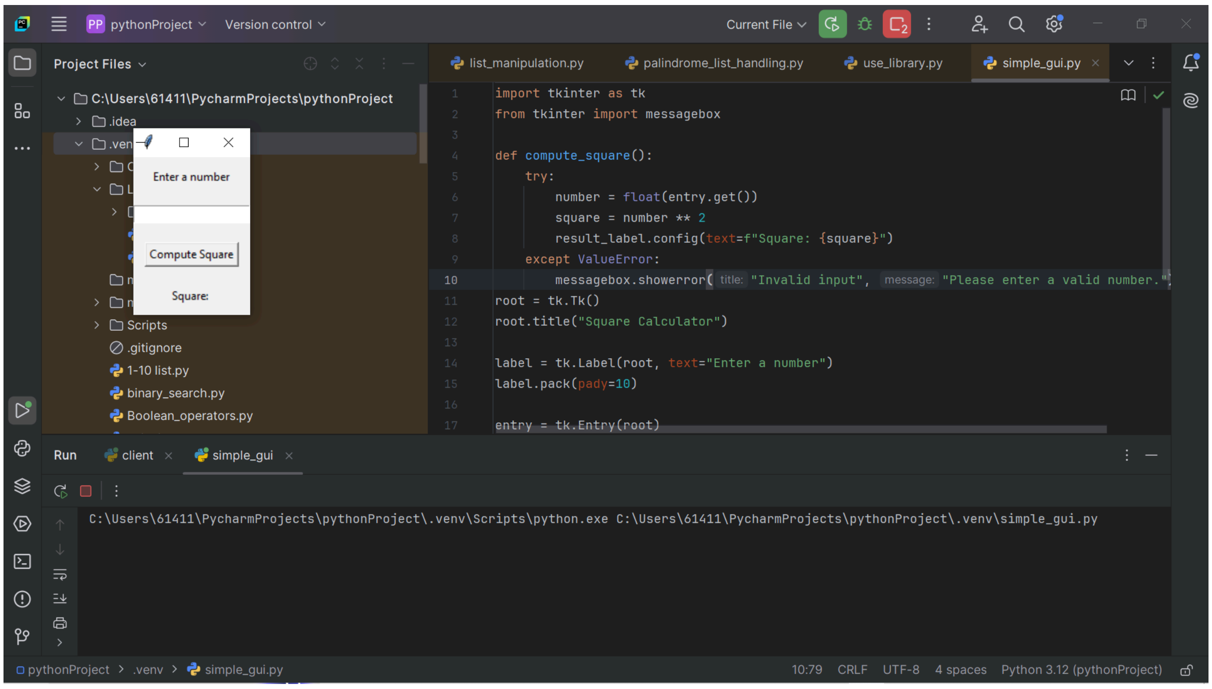
An entry widget to input a number.

A button that, when clicked, computes and displays the square of the entered number.

Code:

```python
import tkinter as tk
from tkinter import messagebox

def compute_square():
   try:
       number = float(entry.get())
       square = number ** 2
       result_label.config(text=f"Square: {square}")
   except ValueError:
       messagebox.showerror("Invalid input", "Please enter a valid number.")
root = tk.Tk()
root.title("Square Calculator")

label = tk.Label(root, text="Enter a number")
label.pack(pady=10)

entry = tk.Entry(root)
entry.pack(pady=10)

compute_button = tk.Button(root, text="Compute Square",
command=compute_square)
compute_button.pack(pady=10)

result_label = tk.Label(root, text="Square: ")
result_label.pack(pady=10)

root.mainloop()
```

Output:



```python
import tkinter as tk
from tkinter import messagebox

def compute_square():
    try:
        number = float(entry.get())
        square = number ** 2
        result_label.config(text=f"Square: {square}")
    except ValueError:
        messagebox.showerror( title: "Invalid input",  message: "Please enter a valid number."
root = tk.Tk()
root.title("Square Calculator")

label = tk.Label(root, text="Enter a number")
label.pack(pady=10)

entry = tk.Entry(root)
```

```
C:\Users\61411\PycharmProjects\pythonProject\.venv\Scripts\python.exe C:\Users\61411\PycharmProjects\pythonProject\.venv\simple_gui.py
```