UNIVERSITY OF ENGINEERING AND TECHNOLOGY, TAXILA

FACULTY OF TELECOMMUNICATION AND INFORMATION ENGINEERING

COMPUTER ENGINEERING DEPARTMENT

Lab 1: Python and Jupyter Basics

Objective

In this lab, students will familiarize themselves with Python programming and the Jupyter Notebook environment. They will learn to:

- Write Python scripts involving variables, loops, and functions.
- Perform basic file handling operations such as reading and writing CSV/JSON files.
- Gain an understanding of the Jupyter Notebook interface and its features.

Prerequisites

- Python and Jupyter Notebook installed on your system.
- Familiarity with basic programming concepts.

Steps and Activities

1. Setting up the Environment

1. Install Jupyter Notebook using pip if not already installed:

   ```
   pip install notebook
   ```

2. Launch Jupyter Notebook:

   ```
   jupyter notebook
   ```

3. Create a new notebook and rename it to Lab1_Python_Basics.

2. Writing Python Scripts

1. **Declare and initialize variables:**

   ```
   name = "John"
   age = 25
   ```

```
print(f"My name is {name}, and I am {age} years old.")
```

2. **Use loops to iterate over a list:**

```
fruits = ["apple", "banana", "cherry"]
for fruit in fruits:
    print(f"I love {fruit}")
```

3. **Define a basic function:**

```
def greet(name):
    return f"Hello, {name}!"
print(greet("Alice"))
```

3. Basic File Handling

CSV (Comma-Separated Values): A plain text file format used to store tabular data, where each row is a record, and fields are separated by commas.

JSON (JavaScript Object Notation): A lightweight data-interchange format that uses key-value pairs to represent structured data in a human-readable and machine-parseable format.

For CSV Files

1. **Create and Write to a CSV File**

```
import csv

# Data to write (list of lists)
data = [
    ["Name", "Age", "City"],  # Header
    ["John", 25, "New York"],
    ["Anna", 30, "Los Angeles"],
    ["Mike", 35, "Chicago"]
]

# Creating and writing to a CSV file
with open('General.csv', mode='w', newline='') as file:
    writer = csv.writer(file)
    writer.writerows(data)  # Write multiple rows

print("CSV file 'General.csv' created and data written successfully!")
```

2. **Read from a CSV File**

```
Import csv
with open('General.csv', mode='r') as file:
    reader = csv.reader(file)
    for row in reader:
        print(row)  # Print each row
```

3. **Append to a CSV File**

```
Import csv
# Data to append
new_data = ["Sarah", 28, "Houston"]

# Appending to the CSV file
with open('General.csv', mode='a', newline='') as file:
    writer = csv.writer(file)
    writer.writerow(new_data)  # Write a single row

print("New data appended to 'people_list.csv' successfully!")
```

For JSON Files

1. **Create and Write to a JSON File**

```
import json

# Data to write (dictionary)
data = {
    "people": [
        {"name": "John", "age": 25, "city": "New York"},
        {"name": "Anna", "age": 30, "city": "Los Angeles"},
        {"name": "Mike", "age": 35, "city": "Chicago"}
    ]
}

# Creating and writing to a JSON file
with open('people_data.json', mode='w') as file:
    json.dump(data, file, indent=4)  # Write JSON with indentation for readability

print("JSON file 'General.json' created and data written successfully!")
```

2. **Read from a JSON File**

```
Import json
with open('people_data.json', mode='r') as file:
    data = json.load(file)  # Load the data as a Python dictionary

# Print the content of the file
print(data)
```

3. **Append to a JSON File**

```
Import json
# Data to append
new_entry = {"name": "Sarah", "age": 28, "city": "Houston"}

# Reading the current file
with open('people_data.json', mode='r') as file:
    data = json.load(file)

# Append the new data
data["people"].append(new_entry)

# Writing back the updated data
with open('people_data.json', mode='w') as file:
    json.dump(data, file, indent=4)  # Re-write the updated data with indentation

print("New data appended to 'people_data.json' successfully!")
```

4. Overview of Jupyter Notebook

1. **Features of Jupyter Notebook:**

    1. **Cells:** Execute Python code one cell at a time.
    2. **Markdown:** Document your code with text, headings, and links.
    3. **Interactive Outputs:** Visualize plots and data outputs inline.

2. **Add a Markdown cell to describe your code:**

    Example: Looping through a list
    This code demonstrates a basic `for` loop in Python.

3. **Use %magic commands for efficiency:**

    %magic commands in Python are special commands in Jupyter Notebook provided by the IPython kernel to enhance efficiency and productivity, allowing you to perform tasks like timing code execution, viewing current variables, and managing the environment seamlessly.

    ```
    %time sum(range(1000000))
    ```

Deliverables

1. A Jupyter Notebook file (Lab1_Python_Basics.ipynb) containing:
    - Examples of variables, loops, and functions.
    - Code for reading and writing CSV/JSON files.
    - Markdown documentation for each section.

Lab Questions

1. Write a Python script that calculates the factorial of a given number.
2. Create a JSON file containing a list of items, read it, and print its contents.
3. Document the steps to open and execute a notebook in Jupyter.

Submission

Submit the following :

1. The completed Jupyter Notebook file.
2. Screenshots or outputs of the tasks performed in the lab.