



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.

1. Reading a CSV file:

```
import pandas as pd
df = pd.read_csv("sample.csv")
print(df.head())
```

2. Writing to a CSV file:

```
df.to_csv("output.csv", index=False)
```

3. Reading and writing JSON files:

```
import json
data = {"name": "John", "age": 25}
with open("data.json", "w") as file:
    json.dump(data, file)
with open("data.json", "r") as file:
```



```
loaded_data = json.load(file)
print(loaded_data)
```

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:

```
%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.



UNIVERSITY OF ENGINEERING AND TECHNOLOGY, TAXILA

**FACULTY OF TELECOMMUNICATION AND INFORMATION
ENGINEERING**

COMPUTER ENGINEERING DEPARTMENT

Submission

Submit the following :

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