

**Data Visualization with Python Lab(CSL48)**

USN:

Week #: 05

Semester:

Section:

Date:

---

**Instructions:**

- Implement the following programs using python language.

**Topic:** Python Modules & Packages, Working with Files

**Programs:**

1. **a. Using Built-in Modules:**

Write a Python program that uses the `math` module to compute the square root, factorial, and greatest common divisor (GCD) of given numbers.

**b. Exploring the `random` Module:**

Write a Python program using the `random` module to generate a list of 5 random integers between 1 and 100.

2. **a. Using `os` and `sys` Modules:**

Write a Python program that prints the current working directory, lists files in a directory, and prints the Python version using the `os` and `sys` modules.

**b. Using `time` Module:**

Write a Python program that prints the current time, pauses execution for 5 seconds using `sleep()`, and then prints a message.

3. **a Copy File Contents:**

Write a Python program that reads a file and writes its contents to another file.

**b. Count Words in a File:**

Write a Python program that reads a file and counts the number of words in it.

4. **a Reading a File in Binary Mode:**

Write a Python program that reads an image or binary file in binary mode and prints its first 100 bytes.

**b. Writing a File in Binary Mode:**

Write a Python program that opens an image file in binary mode and writes it to a new file.

5. **a. Reading a CSV File:**

Write a Python program that reads a CSV file named `data.csv` and prints its content row by Row.

**b. Writing to a CSV File:**

Write a Python program that writes a list of dictionaries to a CSV file with headers

6. **a. Reading a JSON File:**

Write a Python program that reads a JSON file and prints the content in a structured format

**b. Writing to a JSON File:**

Write a Python program that takes user input and writes it to a JSON file.

7. Write a Python program that tries to open a non-existent file and gracefully handles the `FileNotFoundError`.

# Topic: Python Modules & Packages, Working with Files

---

## 1. Python Modules & Packages

### What is a Module?

- A **module** is a Python file (`.py`) that contains definitions of **functions, variables, classes**, etc.
- Python has:
  - **Built-in Modules** like `math`, `random`, `os`, `sys`
  - **User-defined Modules** (your own Python scripts)

### What is a Package?

- A **package** is a collection of **modules** in directories containing a special `__init__.py` file.
- Helps **organize related modules** together.

### Importing Modules

```
import math
from math import sqrt, factorial
import mymodule as mm
```

---

## 2. Working with Files in Python

### File Modes

Mode	Description
<code>r</code>	Read
<code>w</code>	Write (overwrite)
<code>a</code>	Append
<code>b</code>	Binary mode
<code>r+</code>	Read + Write

### Common File Methods

```
file.read(), file.write(), file.readline(), file.readlines()
```

### Using Context Manager

```
with open('file.txt', 'r') as f:
    data = f.read()
```

---

# Programs

---

## ♦ 1. Using Built-in Modules (math)

```
import math

num1 = 16
num2 = 5
num3 = 30
num4 = 45

print("Square root of", num1, "is", math.sqrt(num1))
print("Factorial of", num2, "is", math.factorial(num2))
print("GCD of", num3, "and", num4, "is", math.gcd(num3, num4))
```

---

## ♦ 2. Exploring the **random** Module

```
import random

random_numbers = [random.randint(1, 100) for _ in range(5)]
print("Random numbers:", random_numbers)
```

---

## ♦ 3a. Using **os** and **sys** Modules

```
import os
import sys

print("Current Working Directory:", os.getcwd())
print("Files in Directory:", os.listdir('.'))
print("Python Version:", sys.version)
```

---

## ♦ 3b. Using **time** Module

```
import time

print("Current time:", time.ctime())
time.sleep(5)
print("5 seconds passed!")
```

---

#### ♦ 4a. Copy File Contents

```
with open("source.txt", "r") as src, open("destination.txt", "w")  
as dest:  
    content = src.read()  
    dest.write(content)
```

---

#### ♦ 4b. Count Words in a File

```
with open("sample.txt", "r") as file:  
    text = file.read()  
    word_count = len(text.split())  
    print("Word Count:", word_count)
```

---

#### ♦ 5a. Reading a File in Binary Mode

```
with open("image.jpg", "rb") as file:  
    data = file.read(100)  
    print("First 100 bytes:", data)
```

---

#### ♦ 5b. Writing a File in Binary Mode

```
with open("image.jpg", "rb") as src, open("copy.jpg", "wb") as  
dest:  
    content = src.read()  
    dest.write(content)
```

---

#### ♦ 6a. Reading a CSV File

```
import csv  
  
with open("data.csv", "r") as file:  
    reader = csv.reader(file)  
    for row in reader:  
        print(row)
```

---

### ♦ 6b. Writing to a CSV File

```
import csv

data = [
    {"Name": "Alice", "Age": 24},
    {"Name": "Bob", "Age": 27}
]

with open("output.csv", "w", newline='') as file:
    writer = csv.DictWriter(file, fieldnames=["Name", "Age"])
    writer.writeheader()
    writer.writerows(data)
```

---

### ♦ 7a. Reading a JSON File

```
import json

with open("data.json", "r") as file:
    data = json.load(file)
    print(json.dumps(data, indent=4))
```

---

### ♦ 7b. Writing to a JSON File

```
import json

name = input("Enter your name: ")
age = input("Enter your age: ")

user_data = {"name": name, "age": age}

with open("user.json", "w") as file:
    json.dump(user_data, file, indent=4)
```

---

### ♦ 8. Handle FileNotFoundError Gracefully

```
try:
    with open("nonexistent.txt", "r") as file:
        print(file.read())
except FileNotFoundError:
    print("The file you are trying to open does not exist.")
```