Laboratory Activity No. 3	
Polymorphism	
Course Code: CPE009	**Program:** BSCPE
Course Title: Object - Oriented Programming	**Date Performed:** September 30, 2024
Section: CPE21S4	**Date Submitted:** September 30, 2024
Name: Don Eleazar T. Fernandez	**Instructor:** Maria Rizette Sayo

1. Objective(s):

This activity aims to familiarize students with the concepts of Polymorphism in Object-Oriented Programming

2. Intended Learning Outcomes (ILOs):

The students should be able to:
2.1 Identify the use of Polymorphism in Object-Oriented Programming
2.2 Implement an Object-Oriented Program that applies Polymorphism

3. Discussion:

Polymorphism is a core principle of Object-Oriented that is also called “method overriding”. Simply stated the principles says that a method can be redefined to have a different behavior in different derived classees.

For an example, consider a **base file reader/writer** class then three derived classes **Text file reader/writer**, **CSV file reader/ writer**, and **JSON file reader/writer**. The base file reader/writer class has the methods: **read**(filepath=””) , **write**(filepath=””). The three derived classes (classes that would inherit from the base class) should have behave differently when their read, write methods are invoked.

CSV stands for **Comma Separated Values** while **JSON** stands for **Javascript Server Object Notation**. These are the standard file formats and structures used by applications and systems to transfer/exchange data between their systems. For example, you may visit this online api http://dummy.restapiexample.com/api/v1/employees (note that the data is fake) but this url provides data that another system can consume and use in their system.

4. Materials and Equipment:

Desktop Computer with Anaconda Python
Windows Operating System

5. Procedure:

Creating the Classes

1. Create a folder named oopfa1<lastname>_lab8
2. Open your IDE in that folder.
3. Create the base FileReaderWriter .py file and Class using the code below:

FileReaderWriter.py > ...

```
class FileReaderWriter():
    def read(self):
        print("This is the default read method")

    def write(self):
        print("This is the default write method")
```

4. Create the CSVFileReaderWriter .py and Class using the code below:

CSVFileReaderWriter.py > ...

```
from FileReaderWriter import FileReaderWriter
import csv

class CSVFileReaderWriter(FileReaderWriter):
    def read(self, filepath):
        with open(filepath, newline='') as csvfile:
            data = csv.reader(csvfile, delimiter=',', quotechar='|')
            for row in data:
                print(row)
            return data

    def write(self, filepath, data):
        with open(filepath, 'w', newline='') as csvfile:
            writer = csv.writer(csvfile, delimiter=',',
                                quotechar='|', quoting=csv.QUOTE_MINIMAL)
            writer.writerow(data)
```

5. Create the JSONFileReaderWriter Class using the code below

JSONFileReaderWriter.py > ...

```python
from FileReaderWriter import FileReaderWriter
import json

class JSONFileReaderWriter(FileReaderWriter):
    def read(self, filepath):
        with open(filepath, "r") as read_file:
            data = json.load(read_file)
            print(data)
            return data

    def write(self, filepath, data):
        with open(filepath, "w") as write_file:
            json.dump(obj=data, fp=write_file)
```

Testing and Observing Polymorphism

1. Create a .csv file named sample.csv with the following content. (you may use the IDE or plain notepad)

sample.csv

```
Apple,Banana,Mango,Orange,Cherry
```

2. Create a .json file named sample.json with the following content. (you may use the IDE or plain notepad).

sample.json

```json
{
    "description":"This is a JSON Sample",
    "accounts": [
        {"id":1,"name":"Jack"},
        {"id":2,"name":"Rose"}
    ]
}
```

3. Create the main.py that will test the functionality of the classes.

main.py

```python
from FileReaderWriter import FileReaderWriter
from CSVFileReaderWriter import CSVFileReaderWriter
from JSONFileReaderWriter import JSONFileReaderWriter

# Test the default class
df = FileReaderWriter()
df.read()
df.write()

# Test the polymoprhed methods
c = CSVFileReaderWriter()
c.read("sample.csv")
c.write(filepath="sample2.csv", data=["Hello","World"])

j = JSONFileReaderWriter()
j.read("sample.json")
j.write(data=['foo', {'bar': ('baz', None, 1.0, 2)}],filepath="sample2.json")
```

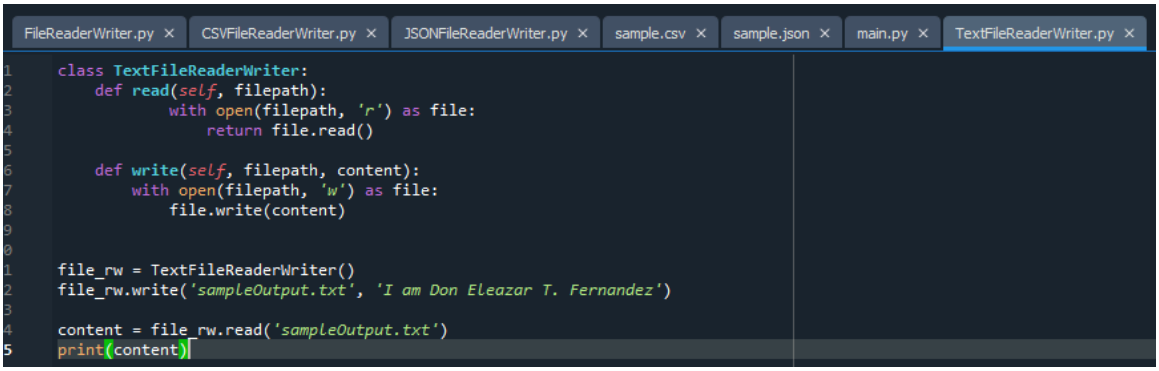
4. Run the program and observe the output carefully the values in sample2.csv and sample2.json.

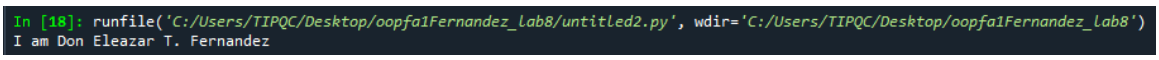
```
This is the default read method
This is the default write method
['Apple', ' Banana', ' Mango', ' Orange', ' Cherry']
{'description': 'This is a JSON Sample', 'accounts': [{'id': 1, 'name': 'Jack'}, {'id': 2, 'name': 'Rose'}]}
```

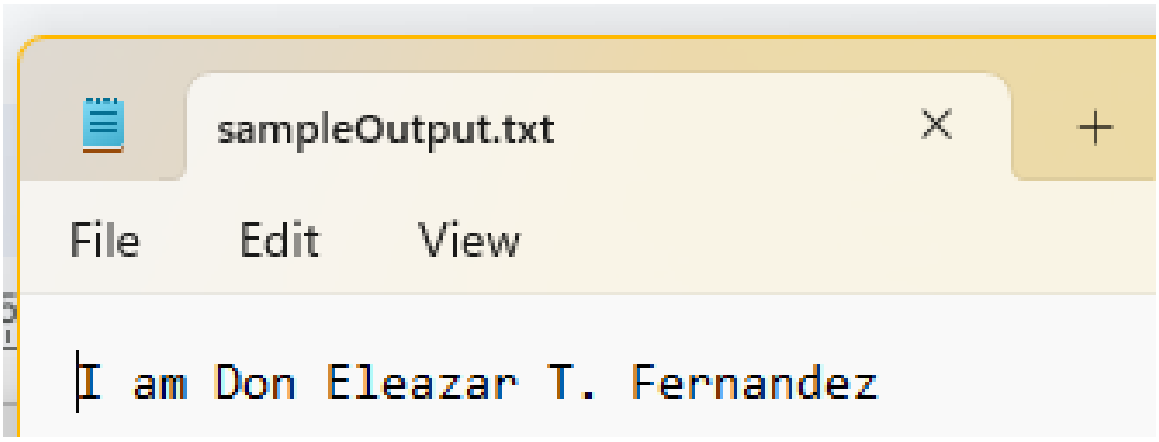
6. Supplementary Activity:

Task

Create a simple TextFileReaderWriter.py file and Class that will be able to **read** from and **write** (override) to a text file. The read and write method should be overridden according to the requirement of Text File Reading and Writing as performed in Laboratory Activity 5.








```
class TextFileReaderWriter:
    def read(self, filepath):
            with open(filepath, 'r') as file:
                return file.read()

    def write(self, filepath, content):
        with open(filepath, 'w') as file:
            file.write(content)


file_rw = TextFileReaderWriter()
file_rw.write('sampleOutput.txt', 'I am Don Eleazar T. Fernandez')

content = file_rw.read('sampleOutput.txt')
print(content)
```

Questions

1. Why is Polymorphism important?
   - In object oriented programming, polymorphism is vital because it makes an object usable as instances of their parent class, this improves the modularity of the code.

2. Explain the advantages and disadvantages of using applying Polymorphism in an Object-Oriented Program.
   - The advantages of polymorphism are code reusability and modularity. While the disadvantage is that it can increase code complexity, making it harder to fix.

3. What maybe the advantage and disadvantage of the program we wrote to read and write csv and json files?
   - With polymorphism, the advantage of the program created is the versatility to apply the program in different file format and ensure at the same time the functionality. The disadvantage is the complicatedness to understand and further develop the structure and functionality of the program.

4. What maybe considered if Polymorphism is to be implemented in an Object-Oriented Program?
   - When implementing polymorphism, one must define a clear structure, an abstract class, and make thorough verification for it to work in different object types effectively.

5. How do you think Polymorphism is used in actual programs that we use today?
   - In this age, polymorphism could be used as the GUI of the program that allows it to be streamed with different elements such as the size, shape, and color.

7. Conclusion:

The ability to take on multiple forms and yet be defined as their parent class is provided by polymorphism. It makes it possible to modify the program to fit various situations. Its benefits include its ability to be maintained, reusability of the program, and adaptability. On the other hand, the program's intricate structure poses a drawback since it may impede its ability to fulfill its intended objective.

8. Assessment Rubric: