Laboratory Activity No. 3	
Polymorphism	
Course Code: CPE009	**Program:** BSCPE
Course Title: Object-Oriented Programming	**Date Performed: 09/30/24**
Section: CPE21S4	**Date Submitted: 09/30/24**
Name: Magistrado, Aira Pauleen M.	**Instructor: Maam 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 > ...

```
1  class FileReaderWriter():
2      def read(self):
3          print("This is the default read method")
4
5      def write(self):
6          print("This is the default write method")
```

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

CSVFileReaderWriter.py > ...

```python
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)
```

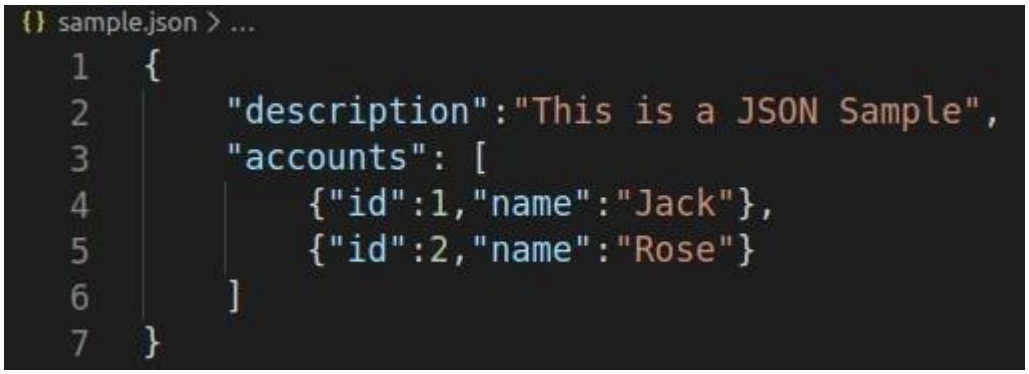
Testing and Observing Polymorphism

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)



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

with o

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

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 overriden according to the requirement of Text File Reading and Writing as performed in Laboratory Activity 5.

CSVFileReaderWriter.py | JSONFileReaderWriter.py | sample.csv | main.py | sample2.jso

```python
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)
```

FileReaderWriter.py | CSVFileReaderWriter.py | JSONFileReaderWriter.py

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

FileReaderWriter.py | CSVFileReaderWriter.py | 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)
```

Writer.py × | CSVFileReaderWriter.py × | JSONFileReaderWriter.py × | sample.csv × | main.py ×

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

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

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

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

t = TextFileReaderWriter()
t.read("sample.text")
t.write(data="Sugar",filepath="sample2.txt")
```

v × | TextFileReaderWriter.py × | sample.text ×

```
1  Blue, VioleT, pURPLE
```

× | sample.text × | sample2.txt ×

```
Sugar
```

TextFileReaderWriter.py × | sample.text × | sample2.txt × | sample

```python
from FileReaderWriter import FileReaderWriter


class TextFileReaderWriter(FileReaderWriter):
    def read(self, filepath):
        with open(filepath, 'r') as txtfile:
            data = txtfile.readlines()
            for line in data:
                print(line.strip())
            return data

    def write(self, filepath, data):
        with open(filepath, 'w') as write_file:
            write_file.write(data)
```

Questions

1. Why is Polymorphism important?

Polymorphism is having the same function names but in many different kind of forms depending on the code's context. Like the subclass allows us to define methods in the parent class with the same name.

2. Explain the advantages and disadvantages of using applying Polymorphism in an Object-Oriented Program.

   Applying Polymorphism enables us to maintain and read the code easier. As the code's can be re-used However, it has some advantage such as it's complexity and difficulty to implement in codes, it may be hard to debug and you may encounter some runtime issues if it is not properly handled.

3. What maybe the advantage and disadvantage of the program we wrote to read and write csv and json files?
   Using csv is ideal for small applications, json files allows for a smaller file size, lightweight and compact and its execution is faster and it is flexible. However the csv is not recommended for large files or large scale projects but json can.

4. What maybe considered if Polymorphism is to be implemented in an Object-Oriented Program?

The things that should be considered when implementing polymorphism is that there are two types mainly the compile time polymorphism where there is method overloading which allows us to define many methods in a class with the same name and run time polymorphism where there is method overriding which allows us to redefine a method in the child class that is already in the parent class.

5. How do you think Polymorphism is used in an actual programs that we use today?
   I think that Polymorphism is integrated in different programs today like in Game development and is implemented in players or enemies and in GUI's.

7. Conclusion:

In the lab activity I have learned about the integration polymorphism in the code. Applying polymorphism helps us maintain the code easier, having easier readability. In the procedure I have learned how to open and read Json and csv files and their differences. Json is for more large projects while csv is just for small applications.

8. Assessment Rubric: