Laboratory Activity No. 3	
Polymorphism	
Course Code: CPE009	**Program:** BSCPE
Course Title: Object-Oriented Programming	**Date Performed:9/30/24**
Section: CPE21S4	**Date Submitted:9/30/24**
Name: Solis, Paul VIncent M.	**Instructor: Prof. 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:

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
```

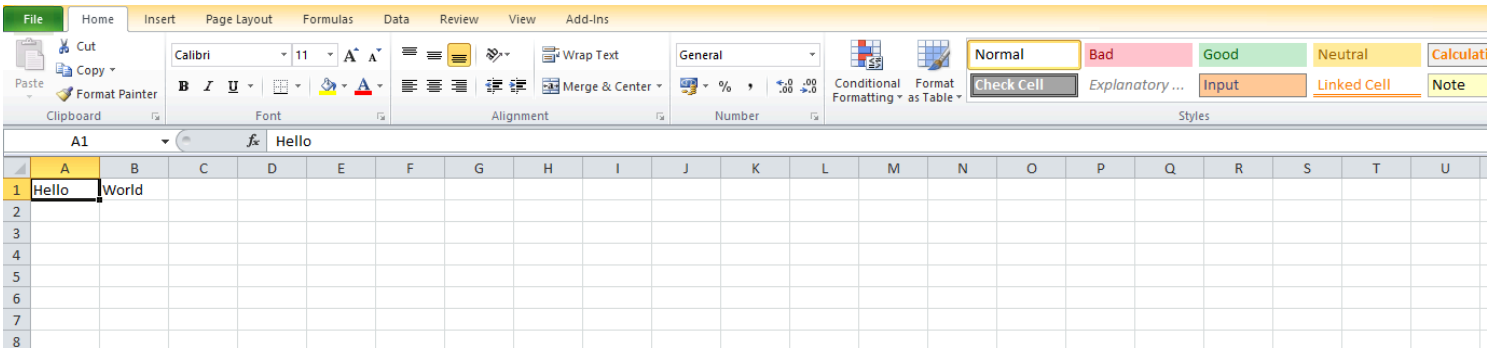
2. Create a .json file named sample.json with the following content. (you may use the IDE or plain notepa

```
{} sample.json > ...
1  {
2      "description":"This is a JSON Sample",
3      "accounts": [
4          {"id":1,"name":"Jack"},
5          {"id":2,"name":"Rose"}
6      ]
7  }
```

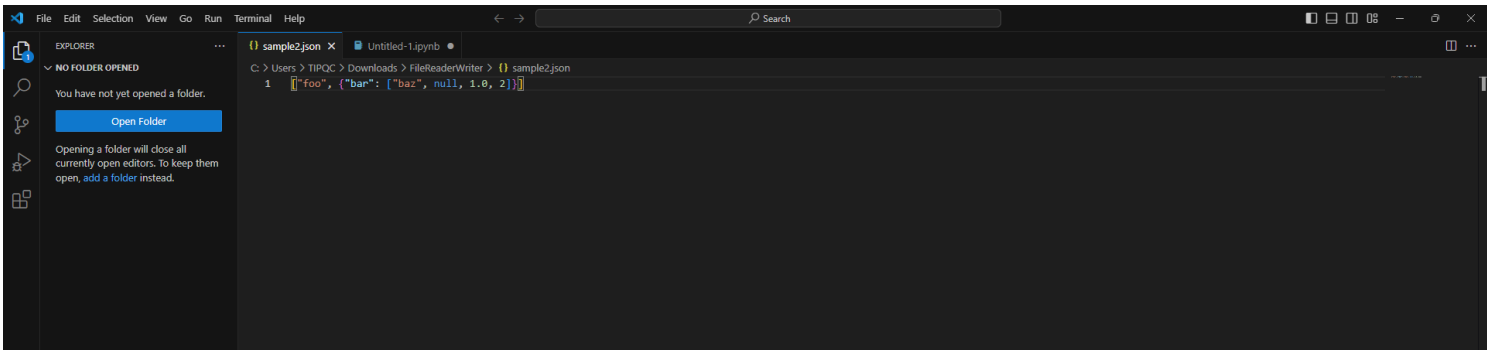
3. Create the main.py that will test the functionality of the classes.

```
main.py > ...
1  from FileReaderWriter import FileReaderWriter
2  from CSVFileReaderWriter import CSVFileReaderWriter
3  from JSONFileReaderWriter import JSONFileReaderWriter
4
5  # Test the default class
6  df = FileReaderWriter()
7  df.read()
8  df.write()
9
10 # Test the polymoprhed methods
11 c = CSVFileReaderWriter()
12 c.read("sample.csv")
13 c.write(filepath="sample2.csv", data=["Hello","World"])
14
15 j = JSONFileReaderWriter()
16 j.read("sample.json")
17 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.



```
In [83]: runfile('C:/Users/TIPQC/Downloads/FileReaderWriter/main.py', wdir='C:/Users/TIPQC/Downloads/FileReaderWrite
Reloaded modules: FileReaderWriter, CSVFileReaderWriter, JSONFileReaderWriter
This is the default read method
This is the default read method
['Apple', ' Banana', ' Mango', ' Orange', ' Cherry']
{'description': 'This is a JSON Sample', 'accounts': [{'id': 1, 'name': 'Jack'}, {'id': 2, 'name': 'Rose'}]}
```

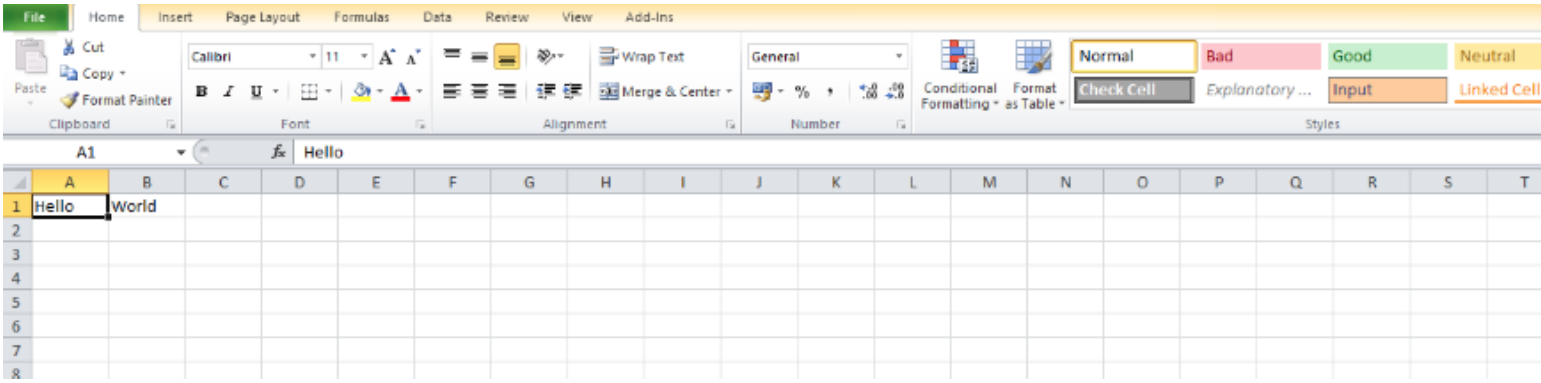


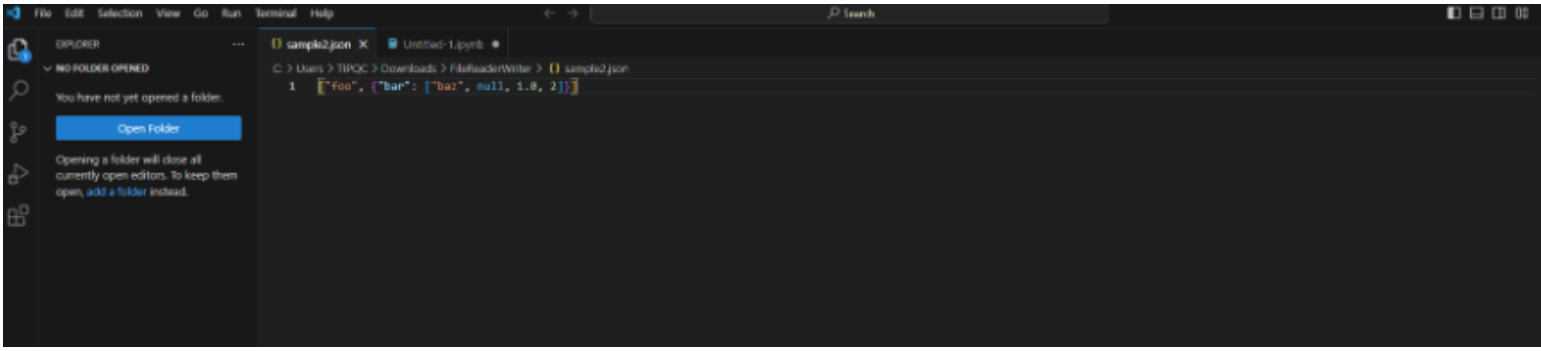
6. Supplementary Activity:

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

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





```python
from FileReaderWriter import FileReaderWriter
from CSVFileReaderWriter import CSVFileReaderWriter
from JSONFileReaderWriter import JSONFileReaderWriter
from TextFileReaderWriter import TextFileReaderWriter  # Import the TextFileReaderWriter

# Test FileReaderWriter (base class)
df = FileReaderWriter()
df.read()
df.write()

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

# Test JSONFileReaderWriter
j = JSONFileReaderWriter()
j.read("sample.json")
# Replace the set with a list
j.write(data=['foo', {'bar': ['baz', None, 1.0, 2]}], filepath="sample2.json")

# Test TextFileReaderWriter
t = TextFileReaderWriter()
t.write(filepath="sample2.txt", data="solis.")
t.read("sample.txt")
```

Questions

1. Why is Polymorphism important?

A key idea in Python that improves code flexibility and reusability is polymorphism. Through a shared interface, it enables objects of different classes to be considered as instances of the same class. Because of this, programmers may create more versatile functions and procedures that work with different kinds of data without having to be aware of their specialized classes. The ability to add new classes without changing the old code makes polymorphism easier to maintain and extend.

Furthermore, this idea encourages improved design and encapsulation by putting more emphasis on an object's capabilities than its limitations. For example, a function that accepts many animal objects can invoke a shared function, such as speak(), irrespective of the animal type. This dynamic capability of Python's polymorphism not only leads to cleaner and more efficient code but also aligns with important software design principles, making it a crucial tool for developers.

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

In object-oriented programming, polymorphism enables many classes to share a common interface and provides benefits like code reusability, flexibility, and easier maintenance. This improves overall code organization by facilitating better abstraction and easier extensions. It may, however, add complexity and performance overhead, making it more difficult for inexperienced developers to comprehend class relationships. Runtime method resolution can also make debugging difficult, and overuse can result in unduly abstract designs that make the architecture more complex. It is essential to weigh these advantages and disadvantages for successful adoption.

3. What maybe the advantage and disadvantage of the program we wrote to read and write csv and json files?

With support for both simple and sophisticated data formats, the application for reading and writing CSV and JSON files has several benefits, such as data interoperability and user-friendliness. It has the advantage of strong libraries that simplify implementation. It does, however, have certain drawbacks, such as the restricted data types that CSV files may support, which might result in data loss or misunderstanding. Furthermore, dealing with very large datasets

may cause performance issues and runtime errors when managing erroneous files. In summary, although the program has its uses, these limitations must be carefully considered.

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

It's crucial to take the structure of your class hierarchy into account when adding polymorphism to an object-oriented software in order to guarantee that related classes have a similar base class or interface. Class relationships can be made more comprehensible with the use of naming rules and clear documentation. In order to prevent any slowdowns, it is also important to evaluate the performance consequences, particularly in large systems. Finally, be sure that polymorphism is used to improve code clarity and maintainability rather than to make it more difficult to understand.

5. How do you think Polymorphism is used in an actual programs that we use today?

Applications like graphical user interfaces (GUIs), where many UI elements (buttons, text fields, etc.) can be handled as instances of a common class, frequently use polymorphism. It is now possible to handle events consistently, which simplifies the management of user interactions. Polymorphism makes it possible for different models to be processed using shared methods in frameworks such as Flask and Django, which streamlines database operations. Furthermore, polymorphism improves code organization and reuse in games by enabling many character types to use an interface for operations like moving and attacking.

7. Conclusion:

In conclusion, polymorphism could be a effective concept in object-oriented programming that upgrades adaptability, reusability, and practicality of code. Its usage permits diverse classes to be treated consistently through shared interfacing, rearranging intuitive in complex frameworks such as GUIs, web systems, and amusement advancement. Whereas it offers noteworthy points of interest, such as advancing cleaner plans and simpler expansions, cautious thought is vital to oversee potential disadvantages like execution overhead and expanded complexity. In general, when connected mindfully, polymorphism can essentially make strides the productivity and clarity of computer program improvement, making it a crucial device in advanced programming hones.

8. Assessment Rubric: