Laboratory Activity No. 3	
Polymorphism	
Course Code: CPE009	**Program:** BSCPE
Course Title: Object-Oriented Programming	**Date Performed: 09-30-2024**
Section: CpE 21s4	**Date Submitted: 09-30-2024**
Name: BONIFACIO, NYKO ADREIN L.	**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:

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)

sample.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

```
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.

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.

TextFileReaderWriter.py

C:\Users\TIPQC\Downloads\New folder\TextFileReaderWriter.py

FileReaderWriter.py × | CSVFileReaderWriter.py × | JSONFileReaderWriter.py × | sample.csv × | sample.json × | main.py × | TextFileReaderWriter.py ×

```python
class TextFileReaderWriter:
    def __init__(self, file_path):
        self.file_path = file_path

    def read(self):
        # Attempt to read the file and handle failure with a message
        with open(self.file_path, 'r') as file:
            return file.read()

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

if __name__ == "__main__":
    reader_writer = TextFileReaderWriter('sample.csv')

    print(reader_writer.write("This is now the new content of sample.csv\n12345, abcde, CPE21S4, OOP009B, TIPQC."))

    try:
        print("File Content:")
        print(reader_writer.read())
    except FileNotFoundError:
        print(f"Error: The file '{reader_writer.file_path}' was not found.")
```

OUTPUT before Overriding

```
In [16]: runfile('C:/Users/TIPQC/Downloads/New folder/main.py', wdir='C:/Users/TIPQC/Downloads/New folder')
Reloaded modules: FileReaderWriter, CSVFileReaderWriter, JSONFileReaderWriter
This is the default read method
This is the defualt write method
['Apple', ' Banana', ' Mango', ' Orange', ' Cherry']
{'description': 'This is a JSON sample', 'accounts': [{'id': 1, 'name': 'Jack'}, {'id': 2, 'name': 'Rose'}]}
```

OUTPUT after Overriding

```
In [29]: runfile('C:/Users/TIPQC/Downloads/New folder/main.py', wdir='C:/Users/TIPQC/Downloads/New folder')
This is the default read method
This is the defualt write method
['12345', ' abcde', ' CPE21S4', ' OOP009B', ' TIPQC.']
{'description': 'This is JSON sample', 'accounts': [{'id': 1, 'name': 'Jack'}, {'id': 2, 'name': 'Rose'}]}
```

Questions

1. Why is Polymorphism important?
    - Polymorphism in OOP is important because it allows objects of different types to be treated uniformly through a common interface, enhancing flexibility, reusability, and scalability. It simplifies code by enabling methods to operate on various object types without needing modification, reduces complexity by eliminating conditional type-checking, and promotes extensibility, allowing new types to be integrated seamlessly.

2. Explain the advantages and disadvantages of using applying Polymorphism in an Object-Oriented Program.
    - Polymorphism offers several advantages in object-oriented programming, such as enhancing code reusability, flexibility, and simplification by allowing a common interface for different object types. However, it can also introduce performance overhead due to dynamic method dispatch, make debugging more complex, and potentially lead to overly generalized or less readable code if misused.

3. What maybe the advantage and disadvantage of the program we wrote to read and write csv and json files?
    - The advantage of the program to read and write CSV and JSON files is that it enables easy data exchange and storage in widely-used formats, making the program versatile and compatible with other systems or applications. However, a disadvantage could be the potential for performance issues with large datasets, as CSV lacks structure and JSON can become memory-intensive, which may slow down processing or increase resource usage.

4. What maybe considered if Polymorphism is to be implemented in an Object-Oriented Program?
    - When implementing polymorphism in an object-oriented program, it is important to consider the program's design for flexibility and scalability, ensuring that a common due to dynamic method dispatch and ensure that the code remains readable and maintainable. Proper planning is essential to avoid overcomplicating the design or misusing polymorphism, which could reduce clarity and increase debugging difficulty.
5. How do you think Polymorphism is used in an actual programs that we use today?
    - Polymorphism is commonly used in modern programs to enhance flexibility and maintainability. For example, in GUIs, components like buttons and sliders are treated uniformly, enabling consistent behavior. It's also applied in frameworks and APIs, allowing different objects to be processed using the same methods, simplifying functionality and communication.interface or base class is used effectively. Developers must also weigh performance costs

7. Conclusion:

8. Assessment Rubric: