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: Aries Rio	**Instructor: Ma'am 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

```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.

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.

task.py > ...

```python
class TextFileReaderWriter:
    def read(self, filepath):
        """Read the contents of a text file and return them as a string."""
        with open(filepath, 'r') as file:
            content = file.read()
        return content

    def write(self, filepath, data):
        """Write data to a text file, overriding its contents."""
        with open(filepath, 'w') as file:
            file.write(data)

# Example usage
if __name__ == "__main__":
    text_file_path = 'example.txt'
    text_writer_reader = TextFileReaderWriter()

    # Writing to the text file
    example_text = "Hello, this is a sample text file.\nIt contains multiple lines."
    text_writer_reader.write(text_file_path, example_text)

    # Reading from the text file
    content = text_writer_reader.read(text_file_path)
    print("Content of the file:")
    print(content)

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\TIPQC\Documents\rio> & C:/Users/TIPQC/AppData/Local/Programs/Python/Python311/python.exe c:/Users/TIPQC/Documents/rio/task.py
Content of the file:
Hello, this is a sample text file.
It contains multiple lines.
PS C:\Users\TIPQC\Documents\rio>
```

Questions

1. Why is Polymorphism important?

   Polymorphism is important in programming because it allows objects of different classes to be treated as if they are the same type through a common interface. This means that methods can be used on different objects without needing to know their exact class. It enhances flexibility and reusability in code, making it easier to manage and extend. By enabling a single function to work with various data types, polymorphism simplifies code and promotes cleaner design.

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

   The advantages of using polymorphism include code reusability, flexibility, and easier maintenance. It allows developers to write code that can work with new classes without modification, which saves time and effort. However, it can also introduce complexity and make debugging more challenging, as understanding the relationships between classes may require more effort. Additionally, polymorphism can add some performance overhead due to dynamic method resolution.

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

   The advantages of the CSV and JSON file program include interoperability, as both formats are widely used for data exchange, and simplicity, making it easy to read and write data. However, disadvantages include limited error handling, which may lead to crashes if files are not found or are in the wrong format. Additionally, the program may struggle with large files, as loading everything into memory can lead to performance issues.

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

   When implementing polymorphism in object-oriented programming, it's important to design clear interfaces that define expected behaviors. This helps maintain consistency across different classes. Developers should also consider using design patterns, thoroughly test their implementations, and document the relationships and behaviors of classes to ensure clarity and maintainability in the codebase.

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

   ~~Polymorphism is commonly used in real-world applications, such as in graphical u~~ser interface (GUI) frameworks where different types of buttons can be treated as a single type for event handling. It's also utilized in data processing systems, where different data readers (like CSV, JSON, and XML) can implement a common interface, allowing interchangeable usage without changing the underlying logic. This flexibility makes it easier to manage and extend applications.

7. Conclusion:

In conclusion, polymorphism is a key concept in object-oriented programming that improves code flexibility and reusability. While it can introduce some complexity, its benefits often outweigh the downsides, especially in larger applications. By using polymorphism effectively, developers can create cleaner and more adaptable code structures, enhancing overall software quality and maintainability.

8. Assessment Rubric: