| Laboratory Activity<br>No. 3 | |
|---|---|
| **Polymorphism** | |
| **Course Code:** CPE009 | **Program:** BSCPE |
| **Course Title:** Object-Oriented Programming | **Date Performed: 09/30/2024** |
| **Section: CPE21S1** | **Date Submitted: 10/02.2024** |
| **Name: Bautista, Jhon Hendricks T.** | **Instructor: Mrs. 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)

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.

**Questions**

1. Why is Polymorphism important?
   Polymorphism is one of the fundamentals of OOP and play an important role for implementing code reusability. It allows us to have a single variable name in storing multiple data types. It is easier for us to read and understand the details by using a common structure for many objects.

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

   Being one of the core principles of OOP, Polymorphism gives us an advantage in allowing objects to have different types but use a single function for all. This reduces the workload for implementing important code and lessens repetition. This will support method overriding and overloading by allowing the classes to provide specific method or multiple methods to have the same name but different parameters to work. As a result, the program will be easier to maintain because adding or changing classes will require less modification for all.

   Even if there are many advantages, we can still have disadvantages when implementing polymorphism. There is a great barrier of complexity towards beginners to grasp and implement the concepts of polymorphism like having method overriding and overloading in the code. With that, there is a higher chance of errors and misuse of polymorphism bringing more bad effects than good to the development of the program.

3. What maybe the advantage and disadvantage of the program we wrote to read and write csv and json files?
   CVS and JSON file both have their pros and cons. In the program we created, the CSV file was appropriate to use in the first file write since we were handling simple tabular data and easy to create. It was perfect to handle the simple list of fruits but using it for the second input of data would be a disadvantage because of difference in data types and and handling the nested objects like we had in the example. This is why we used the JSON file to handle the second file writing because it can easily support those types of data structures. It can represent different data types without having to worry about misrepresentation. The disadvantage can be a slower read and write operations than CSV files and as well as lacking the support of spreadsheet applications since we need additional tool to view JSON files.

4. What maybe considered if Polymorphism is to be implemented in an Object-Oriented Program?
   In applying Polymorphism in your code when we have to juggle many related objects. When that happens a common interface or base class maybe create in order to handle the objects but we can still provide specific functionalities for each object. We will be able to establish a proper type of hierarchy to identify the base class and derived class.

5. How do you think Polymorphism is used in an actual programs that we use today?
   Today, Polymorphism is used in different programs like we frameworks by allowing handlers to have common methods and call them accordingly. Also it can be seen in GUI frameworks where components have common inheritance in a base class allowing different components to use a unified interface. There are also methods of use in the creation of database management systems for clearly defining a class for standard methods.

**7. Conclusion:**

After this laboratory, I learned the concept of polymorphism wherein we can create a common method or function for classes to use. This method can be changed to cater the needs for each class but still follow a common interface. This is one of the important principles in OOP. I can conclude that it can provide many advantages in developing your code because it promotes code reusability and organization. I also learned how to handle and create CSV and JSON files in my programs. I learned that they both have their own uses for data storage and they can help our programs have a reliable storage of data.

**8. Assessment Rubric:**