Activity No. 3	
Polymorphism	
Course Code: CPE009	**Program:** BS Computer Engineering
Course Title: Object-Oriented Programming 2	**Date Performed:** 09 / 30 / 2024
Section: CPE21S4	**Date Submitted:** 09 / 30 / 2024
Name(s): ROALLOS, Jean Gabriel Vincent G.	**Instructor:** Prof. Maria Rizette Sayo
Documentation	

C:\Users\TIPQC\Desktop\OOPFA1-Roallos_lab8\FileReaderWriter.py

FileReaderWriter.py × CSVFileReaderWriter.py × JSONFileReaderW

```
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")
7
8
```

C:\Users\TIPQC\Desktop\OOPFA1-Roallos_lab8\CSVFileReaderWriter.py

FileReaderWriter.py × CSVFileReaderWriter.py × JSONFileReaderWriter.py × sample.csv × sa

```
1   from FileReaderWriter import FileReaderWriter
2   import csv
3
4   class CSVFileReaderWriter(FileReaderWriter):
5       def read(self, filepath):
6           with open(filepath, newline='') as csvfile:
7               data = csv.reader(csvfile, delimiter=',', quotechar='|')
8               for row in data:
9                   print(row)
10              return data
11
12      def write(self, filepath, data):
13          with open(filepath, 'w', newline='') as csvfile:
14              writer = csv.writer(csvfile, delimiter=',',
15                                  quotechar='|', quoting=csv.QUOTE_MINIMAL)
16              writer.writerow(data)
17
18
```

C:\Users\TIPQC\Desktop\OOPFA1-Roallos_lab8\JSONFileReaderWriter.py

FileReaderWriter.py × CSVFileReaderWriter.py × JSONFileReade

```
1   from FileReaderWriter import FileReaderWriter
2   import json
3
4   class JSONFileReaderWriter(FileReaderWriter):
5       def read(self, filepath):
6           with open(filepath, "r") as read_file:
7               data = json.load(read_file)
8               print(data)
9               return data
10
11      def write(self, filepath, data):
12          with open(filepath, 'w') as write_file:
13              json.dump(obj=data, fp=write_file)
14
15
```

C:\Users\TIPQC\Desktop\OOPFA1-Roallos_lab8\sample.csv

FileReaderWriter.py × CSVFileReaderWriter.py × JSONFileReaderWriter.py × sample.csv ×

```
1  Apple,Banana,Mango,Orange,Cherry
2
```

C:\Users\TIPQC\Desktop\OOPFA1-Roallos_lab8\sample.json

FileReaderWriter.py × CSVFileReaderWriter.py × JSONFileReaderWriter.py × sample.csv × sample.json ×

```
1  {
2   "description": "This is a JSON sample",
3   "accounts": [
4       {"id":1, "name":"Jack"},
5       {"id":2, "name":"Rose"}
6       ]
7  }
8
```

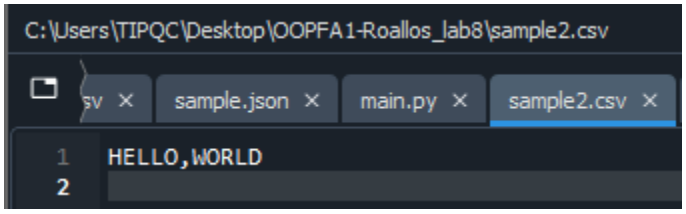
C:\Users\TIPQC\Desktop\OOPFA1-Roallos_lab8\main.py

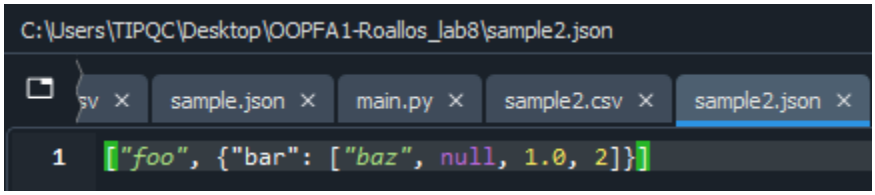
r.py × CSVFileReaderWriter.py × JSONFileReaderWriter.py × sample.csv × sample.json × main.py ×

```
1   from FileReaderWriter import FileReaderWriter
2   from CSVFileReaderWriter import CSVFileReaderWriter
3   from JSONFileReaderWriter import JSONFileReaderWriter
4
5   # Test default class
6   df = FileReaderWriter()
7   df.read()
8   df.write()
9
10  # Test polymorphed 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")
18
```

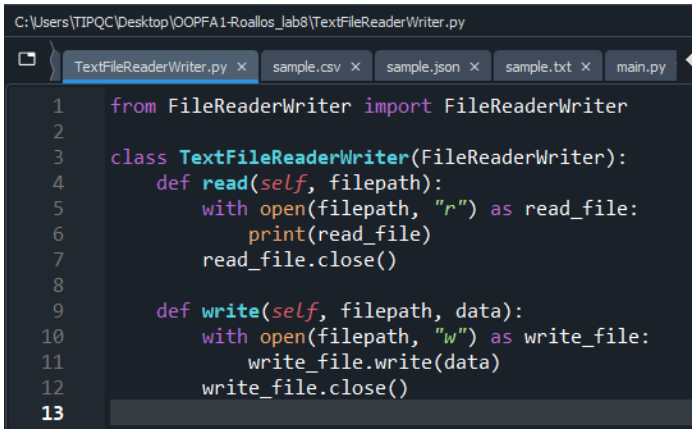
Output

```
In [2]: runfile('C:/Users/TIPQC/Desktop/OOPFA1-Roallos_lab8/main.py', wdir='C:/Users/TIPQC/Desktop/OOPFA1-Roallos_lab8')
Reloaded modules: FileReaderWriter, CSVFileReaderWriter, JSONFileReaderWriter
This is the default read method
This is the default write method
['Apple', 'Banana', 'Mango', 'Orange', 'Cherry']
{'description': 'This is a JSON sample', 'accounts': [{'id': 1, 'name': 'Jack'}, {'id': 2, 'name': 'Rose'}]}

In [3]: runfile('C:/Users/TIPQC/Desktop/OOPFA1-Roallos_lab8/main.py', wdir='C:/Users/TIPQC/Desktop/OOPFA1-Roallos_lab8')
Reloaded modules: FileReaderWriter, CSVFileReaderWriter, JSONFileReaderWriter
This is the default read method
This is the default write method
['Apple', 'Banana', 'Mango', 'Orange', 'Cherry']
{'description': 'This is a JSON sample', 'accounts': [{'id': 1, 'name': 'Jack'}, {'id': 2, 'name': 'Rose'}]}

In [4]: runfile('C:/Users/TIPQC/Desktop/OOPFA1-Roallos_lab8/main.py', wdir='C:/Users/TIPQC/Desktop/OOPFA1-Roallos_lab8')
Reloaded modules: FileReaderWriter, CSVFileReaderWriter, JSONFileReaderWriter
This is the default read method
This is the default write method
['Apple', 'Banana', 'Mango', 'Orange', 'Cherry']
{'description': 'This is a JSON sample', 'accounts': [{'id': 1, 'name': 'Jack'}, {'id': 2, 'name': 'Rose'}]}
```





6. Supplementary Activity



C:\Users\TIPQC\Desktop\OOPFA1-Roallos_lab8\main.py

Writer.py × | JSONFileReaderWriter.py × | TextFileReaderWriter.py × | sample.csv × | sample.json × | sample.txt × | main.py ×

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

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

# Test polymorphed 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")

# Test supplementary activity
t = TextFileReaderWriter()
t.read("sample.txt")
t.write(filepath="sample2.txt", data="This changes the first line!")

```

```
In [10]: runfile('C:/Users/TIPQC/Desktop/OOPFA1-Roallos_lab8/main.py', wdir='C:/Users/TIPQC/Desktop/
OOPFA1-Roallos_lab8')
Reloaded modules: FileReaderWriter, CSVFileReaderWriter, JSONFileReaderWriter, TextFileReaderWriter
This is the default read method
This is the default write method
['Apple', 'Banana', 'Mango', 'Orange', 'Cherry']
{'description': 'This is a JSON sample', 'accounts': [{'id': 1, 'name': 'Jack'}, {'id': 2, 'name':
'Rose'}]}
Hello World!
This is a new line!
This is another new line!

In [11]: runfile('C:/Users/TIPQC/Desktop/OOPFA1-Roallos_lab8/main.py', wdir='C:/Users/TIPQC/Desktop/
OOPFA1-Roallos_lab8')
Reloaded modules: FileReaderWriter, CSVFileReaderWriter, JSONFileReaderWriter, TextFileReaderWriter
This is the default read method
This is the default write method
['Apple', 'Banana', 'Mango', 'Orange', 'Cherry']
{'description': 'This is a JSON sample', 'accounts': [{'id': 1, 'name': 'Jack'}, {'id': 2, 'name':
'Rose'}]}
Hello World!
This is a new line!
This is another new line!
```

Questions

1. Why is Polymorphism important?

   Polymorphism gives code flexibility and reusability. Processes in a code might be used multiple times, in different places and in different data types. With the help of polymorphism, coders can use the process multiple times without writing the same code snippet. This simplifies the coding experience and streamlines productivity.

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

   The code can be more readable and easier to navigate to, as processes can be found in one class or place. This also makes the code process flexible, being able to use it in different places. One disadvantage may be the increased complexity of the code which would not be welcoming for beginners.

3. What may be the advantages and disadvantages of the program we wrote to read and write csv and json files?

   The program can open and manipulate data of files within different types of file formats. Few downsides may be the unintuitive and seemingly primitive editing process for the files and its limitations of what can be edited within the file.

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

   The processes involved for the expected object or objects' attributes are repetitive and needed in various parts of the program. But in implementing polymorphism, the performance of the program should also be considered.

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

   Polymorphism may be used in user interfaces and video game development. Buttons in apps or websites have a general shape and function, and these traits can be contained within the class. In video game development, traits or behavior of an entity within the game might be shared with other entities that might have unique traits or behaviors of its own.

Conclusion:

In summary, the activity shows the use and possible behaviors in polymorphism, such as its clearer code readability and reusability of different processes and behaviors. It simplifies the code maintenance and readability. Along with object encapsulation, it also makes it easier to classify and group traits of entities.