Activity No. 3.1	
Hands-on Activity 3.1 Linked Lists	
Course Code: CPE009B	**Program:** Computer Engineering
Course Title: Object Oriented Programming	**Date Performed:** 09/30/24
Section: CPE21s4	**Date Submitted:** 09/30/24
Name(s): Zolina, Anna Marie	**Instructor:** Mrs. Maria Rizette Sayo

5. Procedure

FileReaderWriter.py | CSVFileReaderWriter.py | JSONFileReaderWriter.

```python
class FileReaderWriter():
    def read(self):
        print("This is the default read method")

    def write(self):
        print("This is the default write method")
```

FileReaderWriter.py | CSVFileReaderWriter.py | JSONFileReaderWriter.py | sample.csv | sample.jsor

```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)
```

```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)
```

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

```
{
    "description" : "This is a JSON Sample",
    "accounts" :
    [
        {"id" : 1, "name":"Jack"},
        {"id" : 2, "name":"Rose"}
    ]
}
```

```
from FileReaderWriter import FileReaderWriter
from CSVFileReaderWriter import CSVFileReaderWriter
from JSONFileReaderWriter import JSONFileReaderWriter

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

# Test the polymorphed objects
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, )}], filepath="sample2.json")
```

7. Supplementary Activity

Task:

TextFileReaderWriter.py:

```
from FileReaderWriter import FileReaderWriter


class TextFileReaderWriter(FileReaderWriter):
    def read(self, filepath):
        with open(filepath, newline = '') as read_file:
            read_file.read()


    def write(self, filepath, data):
        with open(filepath, 'w', newline = '') as write_file:
            write_file.write(data)

```

main.py:

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

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

# Test the polymorphed objects
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, )}], filepath="sample2.json")

tf = TextFileReaderWriter()
tf.read("TextFileReaderWriter.txt")
tf.write(filepath = "New File.txt", data = ("This is the data"))
```

TextFileReaderWriter.txt:

```
Good Morning! This is the new file
```

New File.txt:

```
This is the data
```

Questions:

1. Why is Polymorphism important?

- It is important because the polymorphism is considered to be the one to allow the objects to be instances of the main class. It enhances the main flexibility of the code across its different classes that can be use to reuse and avoid redundancy.

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

- The advantages of Polymorphism is that it can be used multiple times as functions can operate on objects of different classes. It also allows the user or programmer to maintain their system with easier recognition of bugs, and it increases readability to make the code clean and understandable.

- The disadvantages of Polymorphism is that it can be a bit complex for new programmers as the abstract of the given code makes it kind of hard to follow. It can also be a bit confusing since the names of the given classes are the same for almost all of the .py file.

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

- The advantages of Reading and Writing to CSV and JSON FIles is that it is more flexible to handle the different data structures of polymorphism, and the formats are more readable.

- The disadvantages of Reading and Writing to CSV and JSON Files is that there might be data sets that are a bit large for the CSV and JSON that consumes a lot of memory and processing time.

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

- The implementation of the code can be utilized in design patterns and class hierarchies.

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

- It can be used to reducing of code duplication, enhancing modularity, and facilitating extensibility

8. Conclusion

To sum up, there are a lot of benefits, like simplicity of use, flexibility, and interoperability, to having a program that can read and write CSV and JSON files. Developers must be mindful of the drawbacks, though, including possible problems with data integrity, restrictions on performance, and implementation complexity. Through careful consideration of these variables, programmers can easily design strong solutions that take use of these formats' advantages while minimizing their disadvantages.

9. Assessment Rubric