NAME: SANCHEZ, CHRISTAN RAY R.	
COURSE CODE/SECTION: CPE 009B/CPE21S4	
5. Procedure	
SOURCE CODE	

FileReaderWriter

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

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

CSVFileReaderWriter

```
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)
```

JSONFileReaderWriter

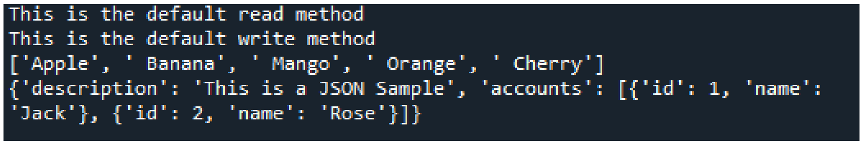
```
from FileReaderWriter import FileReaderWriter
import json

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)
```

csv file

```
{
```

	"description": "This is a JSON Sample", "accounts":[{"id":1,"name":"Jack"}, {"id":2,"name":"Rose"}] }
json file	Apple, Banana, Mango, Orange, Cherry

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 polymorped 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")

OUTPUT	 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:

SOURCE CODE

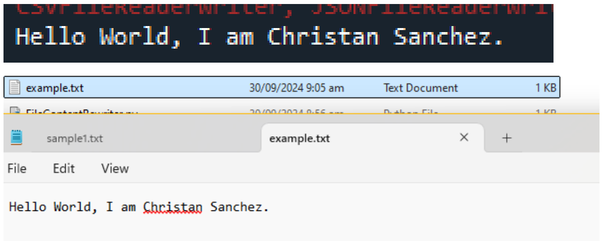
TextFileReaderWriter

```
# TextFileReaderWriter.py

class TextFileReaderWriter:
    def read(self, filepath):
        """Reads the content of a text file and returns it as a string."""
        with open(filepath, 'r', encoding='utf-8') as file:
            content = file.read()
        return content

    def write(self, filepath, data):
        """Writes the given data to a text file, overwriting any existing content."""
        with open(filepath, 'w', encoding='utf-8') as file:
            file.write(data)
```

FileContenRewriter

```
# Example usage
from TextFileReaderWriter import TextFileReaderWriter

# Create an instance of TextFileReaderWriter
file_rw = TextFileReaderWriter()

# Write data to a text file
file_rw.write('example.txt', 'Hello World, I am Christan Sanchez.')

# Read data from the text file
content = file_rw.read('example.txt')
print(content)
```

OUTPUT



```
CSVFileReaderWriter, JSONFileReaderWri
Hello World, I am Christan Sanchez.
```

```
example.txt        30/09/2024 9:05 am        Text Document        1 KB
```

```
Hello World, I am Christan Sanchez.
```

Questions:

1. Why is Polymorphism important?

Polymorphism is crucial in object-oriented programming because it allows objects to be treated as instances of their parent class, simplifying code maintenance and enhancing flexibility. This enables a single interface to represent different underlying forms, making the code more adaptable and easier to manage.

2. Advantages and Disadvantages of Polymorphism in Object-Oriented Programming

Advantages:

- **Code Reusability**: Promotes the reuse of code through inheritance.
- **Flexibility**: Allows for the implementation of dynamic method binding.

Disadvantages:

- **Complexity:** Can make the code more complex and harder to understand.
- **Performance Overhead**: May introduce a slight performance overhead due to dynamic method resolution.

3. Advantages and Disadvantages of a Program for Reading and Writing CSV and JSON Files

Advantages:

- **Versatility**: Supports multiple data formats, making it adaptable to various needs.
- **Ease of Use**: Simplifies data exchange between different systems.

Disadvantages:

- **Error Handling**: Requires robust error handling to manage different file formats.
- **Performance**: May have performance issues with large files.

4. Considerations for Implementing Polymorphism in an Object-Oriented Program

When implementing polymorphism, it’s essential to ensure the design is clear and understandable to avoid confusion. Additionally, consider the potential performance impact of dynamic method binding to maintain efficiency.

5. Real-World Use of Polymorphism

Polymorphism is widely used in software development, such as in GUI frameworks where different objects like buttons and text fields can be treated uniformly. It is also prevalent in APIs, allowing for flexible and scalable code.

7. CONCLUSION

During this activity, I gained a deeper understanding of file manipulation, particularly with JSON and CSV files. I encountered several challenges, such as typos, missing periods, and spacing issues, which highlighted the importance of attention to detail. Despite these hurdles, the experience was incredibly valuable. It not only enhanced my technical skills but also prepared me for future tasks. This hands-on practice reinforced the significance of precision in coding and data handling, making the learning process both rewarding and essential for my growth**.**