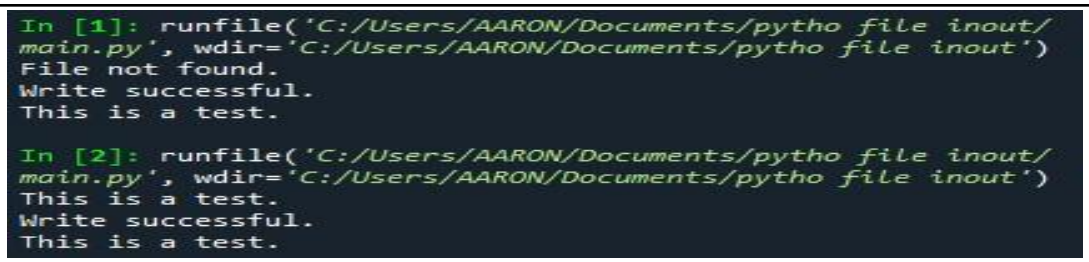
Laboratory Activity no.3	
Polymorphism	
Course Code: CPE009	**Program: BSCPE**
Course Title: OBJECT-ORIENTED PROGRAMING	**Date Performed: SEPTEMBER 26, 2024**
Section: CPE21S1	**Date Submitted: OCTOBER 3, 2024**
Name: GASPAR, AARON ROWEN O.	**Instructor: MAAM MARIA RIZETTE SAYO**

Supplementary Activity:

main.py

```python
# TextFileReaderWriter.py

class TextFileReaderWriter:
    def __init__(self, file_path):
        self.file_path = file_path

    def read_file(self):
        """Reads the contents of the file and returns it as a string."""
        try:
            with open(self.file_path, 'r') as file:
                content = file.read()
            return content
        except FileNotFoundError:
            return "File not found."
        except Exception as e:
            return f"An error occurred: {e}"

    def write_file(self, content):
        """Writes (overrides) the given content to the file."""
        try:
            with open(self.file_path, 'w') as file:
                file.write(content)
            return "Write successful."
        except Exception as e:
            return f"An error occurred: {e}"

file_reader_writer = TextFileReaderWriter('example.txt')

# Read from the file
print(file_reader_writer.read_file())

# Write to the file
print(file_reader_writer.write_file("This is a test."))

# Read from the file again to see the changes
print(file_reader_writer.read_file())
```



Questions:

1. Why is Polymorphism important?

Polymorphism enables the creation of general-purpose functions or methods that can operate on objec of different classes. This means you can write a single function to handle a variety of object types, reducing code duplication and enhancing code reuse. Polymorphic code is also more flexible because allows you to add new classes that adhere to a common interface without modifying existing code. This makes the system easier to extend and maintain, as new functionality can be integrated seamlessly.

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

Polymorphism in object-oriented programming (OOP) offers several notable advantages. It enhances code reusability by allowing developers to write more generic functions and methods that can operate various object types, as long as they adhere to a common interface or superclass. This leads to more flexible and extensible code, where new classes can be added with minimal modifications to existing code. Polymorphism also simplifies code maintenance, as it enables more straightforward and elegant solutions by reducing the need for complex conditional statements. Furthermore, dynamic method bind allows methods to be resolved at runtime based on the actual object type, providing more flexible and adaptable program behavior. Additionally, polymorphism promotes encapsulation and implementation hiding, allowing developers to change the underlying implementation without affecting client code.

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

Creating a program to read from and write to CSV and JSON files offers several advantages and disadvantages. One of the main advantages is data interoperability, as CSV and JSON are widely used formats for data exchange. This ensures compatibility with various applications and systems, facilitating smooth data interoperability. Additionally, the program's flexibility allows it to handle different types of d structures, with CSV being ideal for tabular data and JSON being well-suited for hierarchical data. Another advantage is ease of use, as a simple interface for reading and writing these formats enables users to work with data files without needing to understand the underlying complexities of file handling. Furthermore, automating the process of reading and writing files saves time and reduces the likelihood human error, making the program efficient. Moreover, a well-designed program can be extended to support additional data formats or more complex data processing tasks, enhancing its utility over time.

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

When implementing polymorphism in an object-oriented program, several key considerations must be taken into account. First, it's essential to design a clear and coherent hierarchy of classes, ensuring tha common interfaces or abstract base classes are well-defined. This ensures that derived classes can be

used interchangeably, adhering to the principle of substitutability. Second, careful attention must be pai to method signatures and return types to maintain consistency across different classes, facilitating seamless polymorphic behavior.

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

Polymorphism is a fundamental concept in object-oriented programming that allows objects of different classes to be treated as objects of a common superclass. In actual programs used today, polymorphism commonly employed in scenarios such as GUI frameworks, where different types of UI elements (buttc text boxes, dropdowns) are treated uniformly as generic UI components. This allows developers to writ code that can interact with any type of UI element without needing to know the exact type at compile ti Polymorphism also plays a crucial role in frameworks like Django and Flask, where views can return different types of responses (e.g., HTML, JSON) based on the incoming request, enabling flexible and dynamic handling of various HTTP endpoints. By leveraging polymorphism, modern applications can achieve code reus extensibility, and flexibility in handling diverse data types and behaviors.

Conclusion:

Polymorphism is a fundamental concept in object-oriented programming that allows objects to be treate as instances of their parent class rather than their actual class. This enables a single interface to represent different underlying forms (data types). Understanding and implementing polymorphism can significantly improve your programming skills and lead to more robust and scalable software design.