

Laboratory Activity # 3

Polymorphism

Course Code: CPE009B

Program: BSCPE

Course Title: Object-Oriented Programming

Date Performed: 09-30-2024

Section: CPE21S4

Date Submitted: 09-30-2024

Name(s): Masangkay, Frederick D.

Instructor: Ma'am Maria Rizette Sayo

6. Supplementary Activity

Questions

1. Why is Polymorphism important?
 - Polymorphism is crucial in object-oriented programming as it enables objects from different classes to be treated as instances of a common superclass, promoting code reusability and flexibility. This leads to easier maintenance and scalability, allowing new classes to be added with minimal disruption.
2. Explain the advantages and disadvantages of using applying Polymorphism in an Object-Oriented Program.
 - Polymorphism enhances flexibility and reusability, simplifying code maintenance and allowing easy system extension. However, it can introduce performance overhead, increase complexity, and raise the risk of runtime errors due to type checks occurring at runtime.
3. What maybe the advantage and disadvantage of the program we wrote to read and write csv and json files?
 - A program that handles both CSV and JSON files is flexible and reusable, making it user-friendly. However, this complexity can complicate maintenance and debugging, and managing different formats may lead to bugs and performance issue
 -
4. What maybe considered if Polymorphism is to be implemented in an Object-Oriented Program?
 - When implementing polymorphism in an object-oriented program, several key considerations should be addressed. First, it's essential to design clear and consistent interfaces that define required methods for derived classes, ensuring uniformity. Type safety should be prioritized to minimize runtime errors, and a meaningful inheritance structure must reflect the relationships between classes for better maintainability. Thorough documentation is crucial to help developers understand the design and facilitate future modifications. Additionally, the performance implications of dynamic method resolution should be considered to ensure that flexibility does not compromise efficiency. Finally, robust testing strategies are necessary to verify that polymorphic methods function correctly across various class implementations.
 -
5. How do you think Polymorphism is used in an actual programs that we use today?
 - Polymorphism is essential in modern software applications, enhancing flexibility and functionality. In user interface frameworks, it allows diverse components like buttons and sliders to be managed through a common interface. In game development, different entities can share behaviors while implementing unique functionalities, promoting reusability. APIs use polymorphism for generic functions that handle various data types, while plugin architectures enable loading different extensions through a unified interface. Overall, polymorphism is key to creating adaptable and maintainable software systems.

7. Conclusion

In conclusion, understanding and implementing polymorphism is crucial in object-oriented programming, as it significantly enhances code flexibility, reusability, and maintainability. By allowing different classes to be treated through a common interface, polymorphism simplifies complex systems and facilitates the development of adaptable applications. While it introduces some challenges, such as potential performance overhead and increased complexity, the benefits often outweigh the drawbacks. As demonstrated in real-world applications—from user interfaces to game development and API design—polymorphism is a powerful tool that contributes to efficient and scalable software solutions. Embracing polymorphism ultimately leads to cleaner, more organized code, paving the way for future enhancements and innovations.

8. Assessment Rubric