Explanation of Alignment with Unit Learning Outcomes

Show Wai Yan/105293041

This research project, which compares **Composition vs. Inheritance** in Object-Oriented Programming (OOP), directly aligns with several intended learning outcomes of the unit. By investigating how these two approaches impact software design, the project demonstrates a deep understanding of OOP principles and their practical application, as required for the High Distinction Task 9.3.

The project engages with the unit's learning outcomes in the following ways:

- **Deep Understanding of OOP Concepts**: The research requires a thorough exploration of composition and inheritance as mechanisms for code reuse. By analyzing their theoretical foundations and practical implications, the project showcases a comprehensive grasp of core OOP concepts, such as encapsulation, polymorphism, and modularity.
- **Application of Design Principles**: Designing and implementing two versions of a program—one using composition and one using inheritance—demonstrates the ability to apply OOP principles to solve a practical problem. This hands-on approach highlights how design choices affect software quality, aligning with the unit's emphasis on applying theoretical knowledge.
- **Critical Analysis and Evaluation**: Comparing the two approaches through metrics like maintainability, flexibility, and coupling involves critical thinking about their trade-offs. This analytical process reflects the unit's goal of fostering the ability to evaluate OOP techniques and their impact on software development, ensuring a deep and reflective understanding of the subject matter.

By addressing these learning outcomes, the project not only fulfills the task requirements but also contributes to a broader understanding of effective software design practices in OOP.