KariAnn Harjo

ITSE 1450

Module 6 Discussion 3

Polymorphism is a fundamental concept in object-oriented programming, enabling the same function to perform various actions depending on the object it is acting upon. However, this versatility can sometimes lead to unforeseen consequences when not managed with precision.

One such instance might arise from overriding methods across subclasses. Consider a scenario where different shapes—circles, squares, triangles—are all derived from a single `Shape` class, each with their own version of a `draw()` method. An error in the `Triangle` class that causes the `draw()` method to erase rather than render the shape can result in unexpected deletion of triangles when a uniform action was intended across various shapes. Similarly, automatic type conversion can misdirect a function to an unintended method overload, altering the course of the program in a surprising way.

Dynamic binding, a process where the method to be executed is determined at runtime, can also be a source of surprise. For instance, incorporating a new subclass with a different method behavior into existing code could alter the expected outcomes, as the new subclass's methods might be invoked unexpectedly.

Moreover, confusion between method overloading (methods with the same name but different parameters) and overriding (redefining a method in a subclass) could cause the incorrect method to be executed if the parameters are mismatched, especially in languages that resolve overloading late

Lastly, issues may emerge from interface implementation when a class is forced to implement methods that it doesn't require. These methods may be left with a default or empty implementation, leading to a no-operation result when they are called, defying the anticipation of a meaningful action.

Hence, while polymorphism enhances flexibility, it necessitates vigilant design and thorough understanding to avert these kinds of unexpected outcomes. Proper documentation and careful planning of class hierarchies are essential to leverage the benefits of polymorphism without succumbing to its potential complications.