**Week 1 Assignment**

**Calculate Pay Program**

Eric Vara

The University of Arizona Global Campus

CPT 307: Data Structures & Algorithms

Professor Joel Short

February 2, 2024

**The Role of Operating Systems in Digital Infrastructures**

The Java program described below showcases several key object-oriented programming (OOP) features: encapsulation, abstraction, polymorphism, and, to a lesser extent, inheritance.

A screen shot of a computer

Description automatically generated

Encapsulation is evident in how the Employee class is designed. It bundles the employee-related data (name, rate of pay, hours worked, etc.) and behaviors (setters and getters) into a single unit, the Employee object. This encapsulation hides the internal state of the object and can only be accessed through the methods provided, ensuring data integrity.

Abstraction is achieved through the PayrollCalculator interface, which defines a contract for payroll calculations without specifying the implementation details. This allows for the separation of what needs to be done from how it's done, making the system easier to understand and work with. Concrete classes like SimplePayrollCalculator provide specific implementations of this interface, showcasing the use of polymorphism—the ability to present the same interface for differing underlying forms (implementations). This polymorphism is further demonstrated by the program's ability to use PayrollCalculator references to invoke methods on objects of SimplePayrollCalculator or any other class that implements PayrollCalculator.

Inheritance is implied through the use of interfaces and can be extended to class hierarchies if the program evolves. For example, different types of PayrollCalculators could inherit from a common abstract class that implements the PayrollCalculator interface, sharing common methods while also providing specialized behavior.

Overall, these OOP features help structure the Java program in a way that enhances modularity, maintainability, and scalability. By adhering to these principles, the program not only becomes more organized and flexible but also leverages the full power of Java's object-oriented capabilities.