Laboratory Exercise #8

Polymorphism

Name/s: ***Mirasol, Eger L.***

Section: ***BSCPE 1-5***

Date: ***June 05, 2025***

Instructions: Design a payment processing system that uses polymorphism to handle different types of payment methods.

Scenario:

You're building a system for an e-commerce platform. Customers can pay using various methods: credit card, PayPal, cryptocurrency, etc. Each method has its own way of processing payments, but the system should be able to handle all of them uniformly.

Requirements:

1. **Define an abstract base class or interface PaymentMethod** with:
    - An abstract method process_payment(amount: float)
    - An abstract method validate() (can be overridden for specific validation logic).
2. **Create at least three concrete subclasses:**
    - CreditCardPayment: Validates card number and processes payment.
    - PayPalPayment: Validates email and simulates login before processing.
    - CryptoPayment: Simulates checking a wallet and confirms blockchain transfer.
3. **Create a function checkout(payment: PaymentMethod, amount: float)** that:
    - Calls validate() on the payment method.
    - Calls process_payment(amount) to complete the transaction.
4. **In the main/test function:**
    - Instantiate each payment type with sample data.
    - Call checkout() using different payment objects.
    - Demonstrate that different payment behaviors are triggered using polymorphism.

Expected Concepts Demonstrated:

- Abstract classes/interfaces
- Method overriding
- Polymorphic function calls
- Use of a single interface for varied behaviors