

Home Assignment <14>: Real-World Polymorphism: Payment Gateway Example

Learning Objective:

The objective of this assignment is to understand and apply polymorphism in Python by implementing different payment methods that share the same interface but have different behaviors.

Expected Completion Time:

Best Case: 15 minutes
Average Case: 25 minutes

Assignment Details:

You are building a payment system that supports multiple payment methods.

Requirements:

- a) Create three classes: `CreditCardPayment`, `PayPalPayment`, and `BankTransferPayment`.
- b) In each class, define a method `process_payment(amount)` that prints how the payment is processed. Examples:
 - `CreditCardPayment` → "Processing credit card payment of \$<amount>"
 - `PayPalPayment` → "Processing PayPal payment of \$<amount>"
 - `BankTransferPayment` → "Processing bank transfer of \$<amount>"
- c) Write a function `make_payment(payment_method, amount)` that accepts an object and calls its `process_payment(amount)` method.
- d) In the main section:
 - Create one object of each payment class.
 - Call `make_payment()` with different payment objects to demonstrate polymorphism.

Hints:

1. Each class implements the same method name `process_payment()` but with different logic.
2. Polymorphism is shown when the same function (`make_payment`) can work with different object types.
3. Use a loop to call `make_payment()` for multiple payment objects.

Expected Outcome:

Upon completion of this assignment, you should be able to:

- Implement polymorphism with different classes.
- Use the same method name for different behaviors.
- Write reusable functions that can work with any object implementing a common method.
- Understand how polymorphism applies to real-world domains like payments, transport, etc.