

Task(11.1)

Payment Processing System Using Inheritance in C++

In this task, you will implement a payment processing system using object-oriented programming (OOP) principles, particularly focusing on inheritance in C++. The system will simulate payments made via different methods, such as credit cards and PayPal, using a common base class for shared functionality. The key objective is to demonstrate how derived classes can extend and specialize behavior defined in a base class by overriding methods.

Key Requirements:

1. Base Class: PaymentMethod:

- Define a base class `PaymentMethod` to represent a general payment method. It should contain:
 - A protected data member `amount` to store the payment amount.
 - A constructor to initialize `amount`.
 - A setter `set_amount()` and a getter `get_amount()` for managing the payment amount.

2. Derived Class: CreditCard:

- Create a derived class `CreditCard` that inherits from `PaymentMethod` and represents credit card payments. It should contain:
 - A private data member `cardNumber` to store the credit card number.
 - A constructor that initializes the `amount` and `cardNumber`.
 - A method `processPayment()` that simulates processing a credit card payment by printing out the `amount` and `card number`.

3. Derived Class: PayPal:

- Create another derived class `PayPal` that also inherits from `PaymentMethod` and represents PayPal payments. It should include:
 - A private data member `email` to store the user's PayPal email address.
 - A constructor that initializes the `amount` and `email`.
 - A method `processPayment()` that simulates processing a PayPal payment by printing out the `amount` and the `email address`.

4. Main Program:

- In the `main()` function, create instances of the `CreditCard` and `PayPal` classes.
- Call the `processPayment()` method on each instance to simulate processing a payment using both methods.

Thank You