Create an abstract class PaymentMethod with a pure virtual function processPayment().

Derive classes CreditCardPayment, PayPalPayment, and CashPayment, implementing the processPayment() function differently for each payment method.

#include <iostream>

using namespace std;

// Abstract class

class PaymentMethod {

public:

// Pure virtual function

virtual void processPayment() = 0;

};

// Derived class for Credit Card Payment

class CreditCardPayment : public PaymentMethod {

public:

int p;

void processPayment() {

cout<<"enter credit card payment";

cin>>p;

cout << "Processing credit card payment." << p<<endl;

}

};

// Derived class for PayPal Payment

class PayPalPayment : public PaymentMethod {

public:

int pay;

void processPayment(){

cout<<"enter paypalpayment";

cin>>pay;

cout << "Processing PayPal payment."<<pay << endl;

}

};

// Derived class for Cash Payment

class CashPayment : public PaymentMethod {

public:

int cash;

void processPayment() {

cout<<"enter cashpayment";

cin>>cash;

cout << "Processing cash payment."<<cash << endl;

}

};

// Main function

int main() {

PaymentMethod\* payment;

// Credit card payment

CreditCardPayment creditCard;

payment = &creditCard;

payment->processPayment();

// PayPal payment

PayPalPayment payPal;

payment = &payPal;

payment->processPayment();

// Cash payment

CashPayment cash;

payment = &cash;

payment->processPayment();

return 0;

}