**Question no. 1**

→ **#include <iostream>**

**#include <string>**

**class Address {**

**public:**

**std::string name;**

**std::string address;**

**std::string city;**

**std::string state;**

**std::string zip;**

**Address(std::string name = "", std::string address = "",**

**std::string city = "", std::string state = "", std::string zip = "")**

**: name(name), address(address), city(city), state(state), zip(zip) {}**

**};**

**class Package {**

**protected:**

**Address sender;**

**Address recipient;**

**double weight;**

**double cost\_per\_ounce;**

**public:**

**Package(Address sender, Address recipient, double weight = 0.0,**

**double cost\_per\_ounce = 0.0)**

**: sender(sender), recipient(recipient), weight(weight > 0 ? weight : 0),**

**cost\_per\_ounce(cost\_per\_ounce > 0 ? cost\_per\_ounce : 0) {}**

**virtual double calculateCost() { return weight \* cost\_per\_ounce; }**

**};**

**class TwoDayPackage : public Package {**

**private:**

**double flat\_fee;**

**public:**

**TwoDayPackage(Address sender, Address recipient, double weight = 0.0,**

**double cost\_per\_ounce = 0.0, double flat\_fee = 0.0)**

**: Package(sender, recipient, weight, cost\_per\_ounce),**

**flat\_fee(flat\_fee > 0 ? flat\_fee : 0) {}**

**double calculateCost() override {**

**return Package::calculateCost() + flat\_fee;**

**}**

**};**

**class OvernightPackage : public Package {**

**private:**

**double extra\_fee;**

**public:**

**OvernightPackage(Address sender, Address recipient, double weight = 0.0,**

**double cost\_per\_ounce = 0.0, double extra\_fee = 0.0)**

**: Package(sender, recipient, weight, cost\_per\_ounce),**

**extra\_fee(extra\_fee > 0 ? extra\_fee : 0) {}**

**double calculateCost() override {**

**return weight \* (cost\_per\_ounce + extra\_fee);**

**}**

**};**

**int main() {**

**Address sender("Max Ryder", "1000 San Diario", "San Jose", "CA", "75555");**

**Address recipient("Mark Smith", "7 Elm St", "New York", "NY", "22222");**

**Package standardPackage(sender, recipient, 8.5, 0.5);**

**std::cout << "Standard Shipping Option\n\nSender:\n\n"**

**<< sender.name << "\n"**

**<< sender.address << "\n"**

**<< sender.city << ", " << sender.state << " " << sender.zip**

**<< "\n\nRecipient:\n\n"**

**<< recipient.name << "\n"**

**<< recipient.address << "\n"**

**<< recipient.city << ", " << recipient.state << " " << recipient.zip**

**<< "\n\nCost: $" << standardPackage.calculateCost()**

**<< "\n\n========================================\n\n";**

**TwoDayPackage twoDayPackage(sender, recipient, 8.5, 0.5, 2.0);**

**std::cout << "Two-Day Shipping Option\n\nSender:\n\n"**

**<< sender.name << "\n"**

**<< sender.address << "\n"**

**<< sender.city << ", " << sender.state << " " << sender.zip**

**<< "\n\nRecipient:\n\n"**

**<< recipient.name << "\n"**

**<< recipient.address << "\n"**

**<< recipient.city << ", " << recipient.state << " " << recipient.zip**

**<< "\n\nCost: $" << twoDayPackage.calculateCost()**

**<< "\n\n========================================\n\n";**

**OvernightPackage overnightPackage(sender, recipient, 8.5, 0.5, 0.13);**

**std::cout << "Overnight Shipping Option\n\nSender:\n\n"**

**<< sender.name << "\n"**

**<< sender.address << "\n"**

**<< sender.city << ", " << sender.state << " " << sender.zip**

**<< "\n\nRecipient:\n\n"**

**<< recipient.name << "\n"**

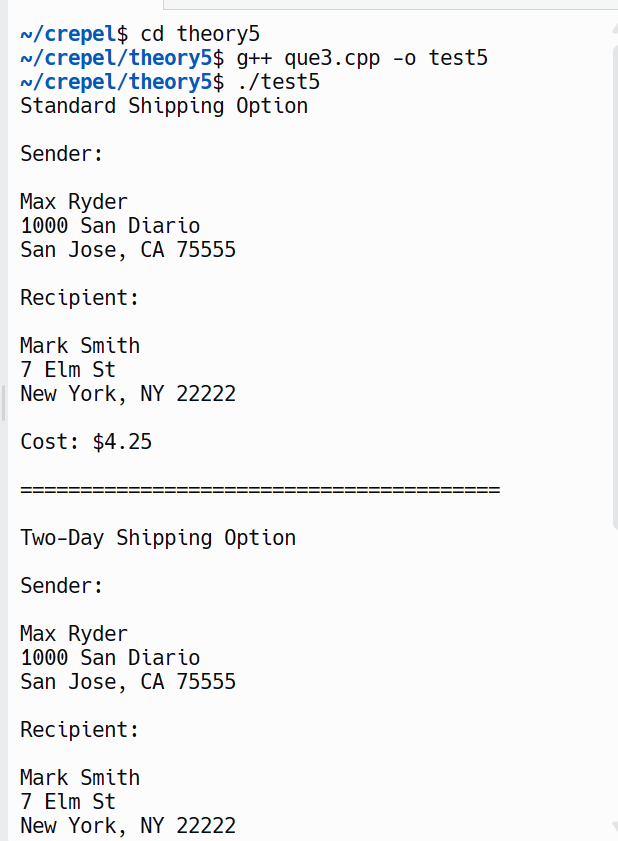
**<< recipient.address << "\n"**

**<< recipient.city << ", " << recipient.state << " " << recipient.zip**

**<< "\n\nCost: $" << overnightPackage.calculateCost();**

**return 0;**

**}**

****

**Question no. 2**

**→ #include <iostream>**

**#include <string>**

**class Product {**

**private:**

**long barcode;**

**std::string name;**

**public:**

**// Constructor with parameters (base and member initializer)**

**Product(long barcode = 0, const std::string &name = "")**

**: barcode(barcode), name(name) {}**

**// Access methods**

**void setCode(long barcode) { this->barcode = barcode; }**

**long getCode() const { return barcode; }**

**// Display product data**

**void scanner() const {**

**std::cout << "Scanning product: " << name << " (Barcode: " << barcode**

**<< ")\n";**

**}**

**void printer() const {**

**std::cout << "Product details: " << name << " (Barcode: " << barcode**

**<< ")\n";**

**}**

**};**

**class PrepackedFood : public Product {**

**private:**

**double unitPrice;**

**public:**

**// Constructor with parameters (base and member initializer)**

**PrepackedFood(long barcode = 0, const std::string &name = "",**

**double unitPrice = 0.0)**

**: Product(barcode, name), unitPrice(unitPrice) {}**

**// Access method for unit price**

**double getUnitPrice() const { return unitPrice; }**

**// Redefine scanner() to include unit price**

**void scanner() const {**

**Product::scanner();**

**std::cout << "Unit price: $" << unitPrice << "\n";**

**}**

**};**

**class FreshFood : public Product {**

**private:**

**double weight;**

**double pricePerKilo;**

**public:**

**// Constructor with parameters (base and member initializer)**

**FreshFood(long barcode = 0, const std::string &name = "", double weight = 0.0,**

**double pricePerKilo = 0.0)**

**: Product(barcode, name), weight(weight), pricePerKilo(pricePerKilo) {}**

**// Access methods for weight and price per kilo**

**double getWeight() const { return weight; }**

**double getPricePerKilo() const { return pricePerKilo; }**

**// Redefine scanner() to include weight and price per kilo**

**void scanner() const {**

**Product::scanner();**

**std::cout << "Weight: " << weight << " kg\n";**

**std::cout << "Price per kilo: $" << pricePerKilo << "\n";**

**}**

**};**

**int main() {**

**// Create objects**

**Product product1(12345, "Apple");**

**PrepackedFood product2(67890, "Chips", 2.99);**

**FreshFood product3(54321, "Banana", 0.5, 1.49);**

**// Test methods**

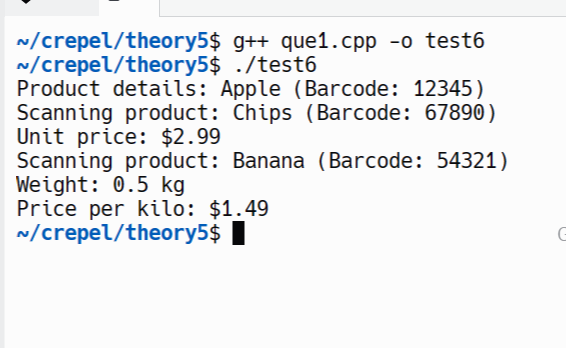
**product1.printer();**

**product2.scanner();**

**product3.scanner();**

**return 0;**

**}**

****