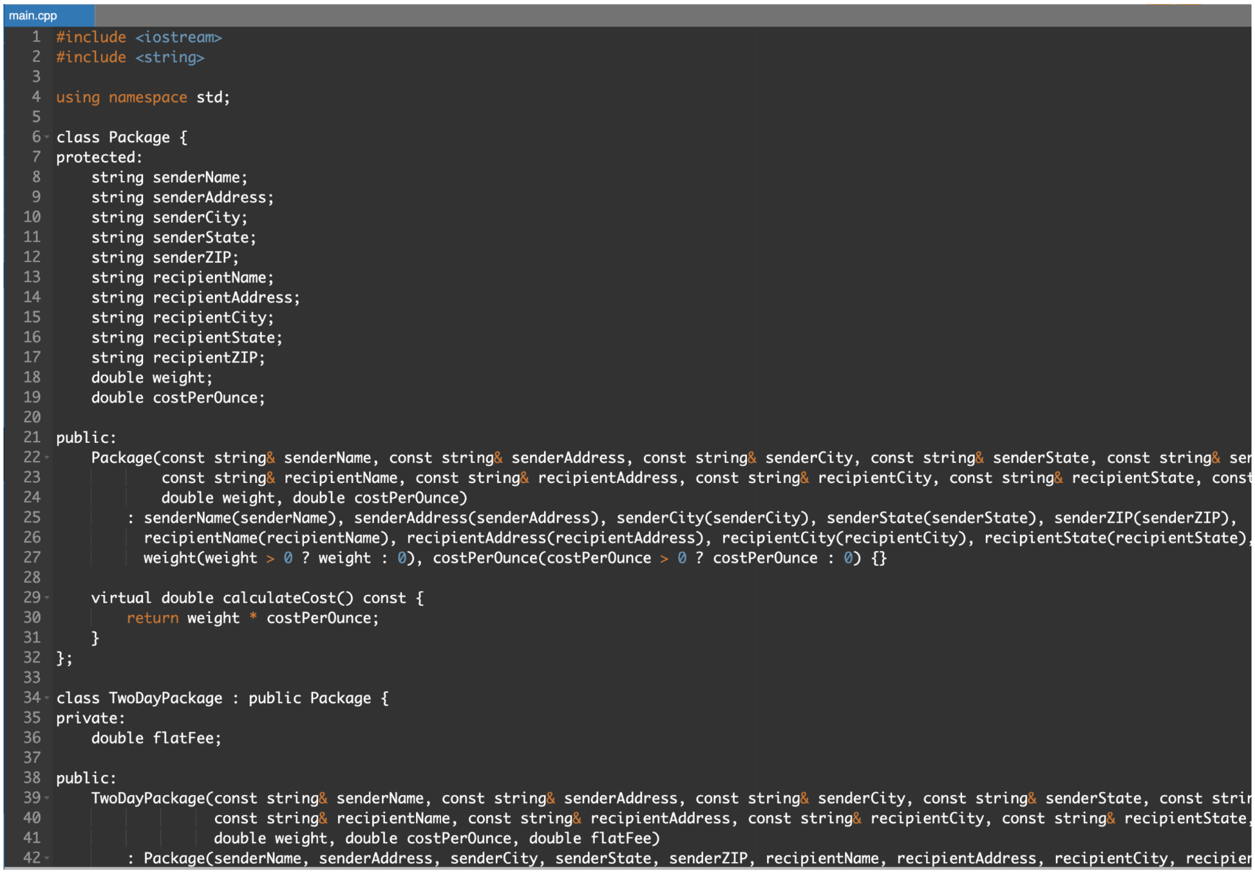
Samir Khadka

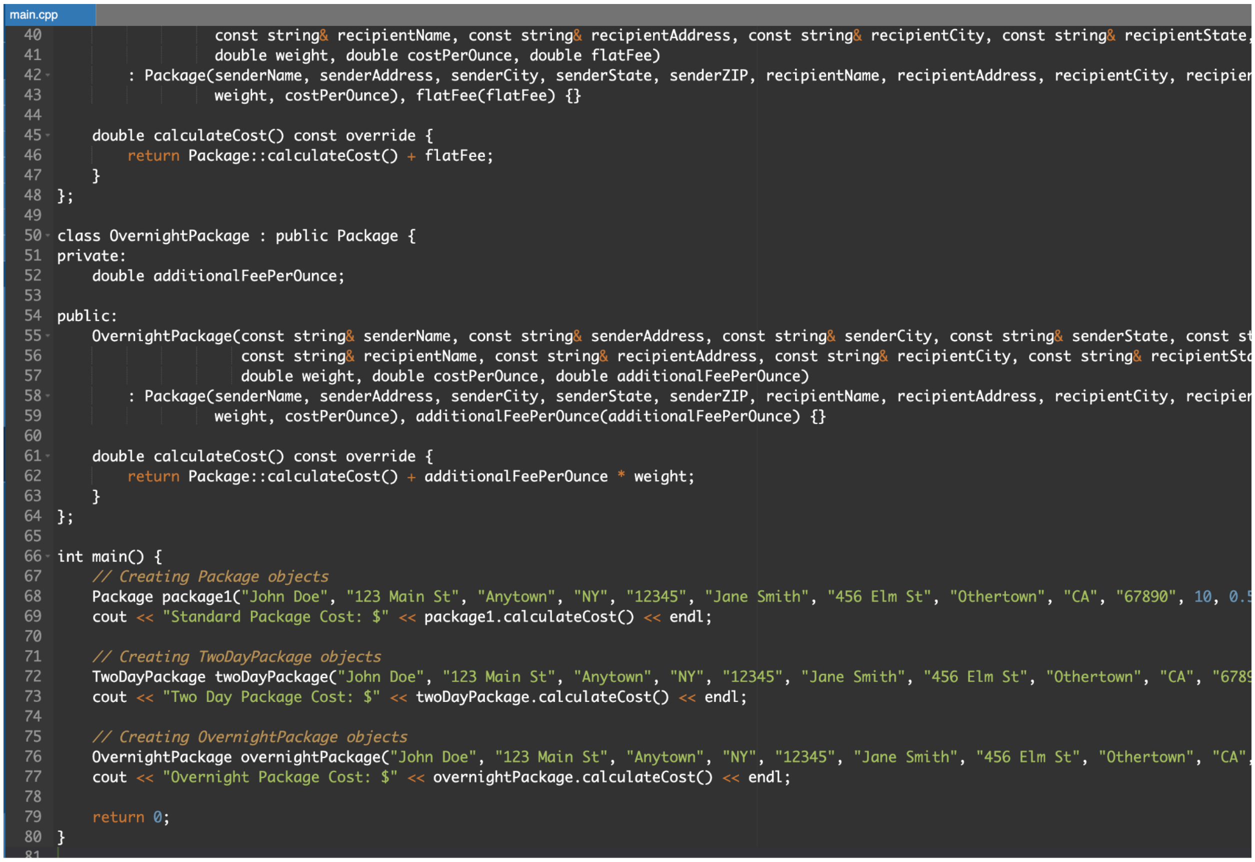
CS360 - Programming in C and C++

Homework Assignment #5

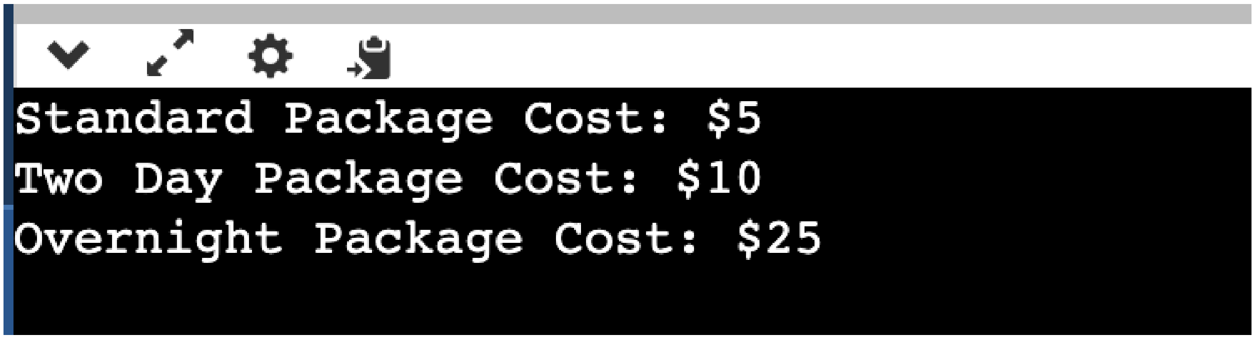
Question 1:



```cpp
#include <iostream>
#include <string>

using namespace std;

class Package {
protected:
    string senderName;
    string senderAddress;
    string senderCity;
    string senderState;
    string senderZIP;
    string recipientName;
    string recipientAddress;
    string recipientCity;
    string recipientState;
    string recipientZIP;
    double weight;
    double costPerOunce;

public:
    Package(const string& senderName, const string& senderAddress, const string& senderCity, const string& senderState, const string& ser
            const string& recipientName, const string& recipientAddress, const string& recipientCity, const string& recipientState, const
            double weight, double costPerOunce)
        : senderName(senderName), senderAddress(senderAddress), senderCity(senderCity), senderState(senderState), senderZIP(senderZIP),
          recipientName(recipientName), recipientAddress(recipientAddress), recipientCity(recipientCity), recipientState(recipientState),
          weight(weight > 0 ? weight : 0), costPerOunce(costPerOunce > 0 ? costPerOunce : 0) {}

    virtual double calculateCost() const {
        return weight * costPerOunce;
    }
};

class TwoDayPackage : public Package {
private:
    double flatFee;

public:
    TwoDayPackage(const string& senderName, const string& senderAddress, const string& senderCity, const string& senderState, const strin
                  const string& recipientName, const string& recipientAddress, const string& recipientCity, const string& recipientState,
                  double weight, double costPerOunce, double flatFee)
        : Package(senderName, senderAddress, senderCity, senderState, senderZIP, recipientName, recipientAddress, recipientCity, recipien
```

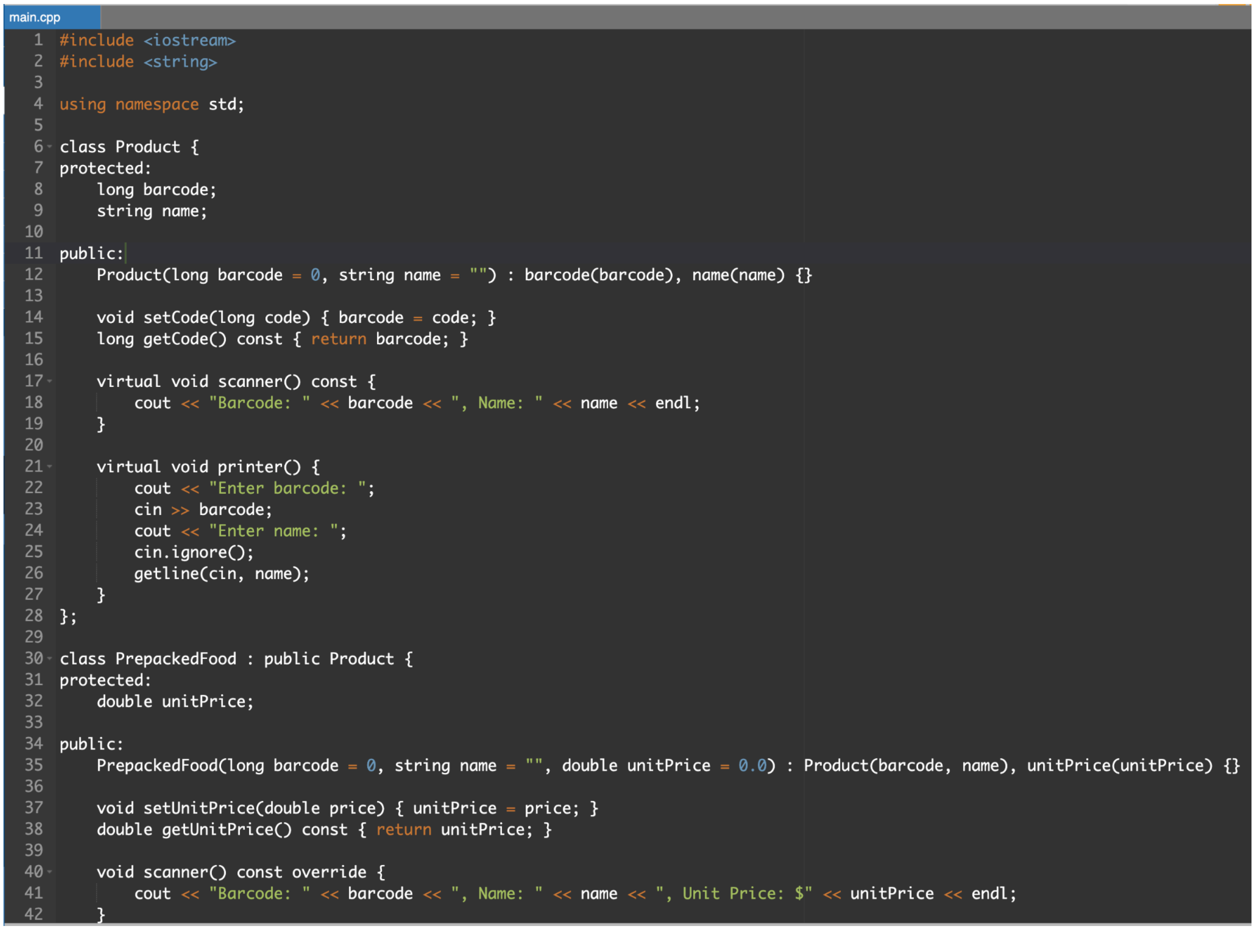


```
                  const string& recipientName, const string& recipientAddress, const string& recipientCity, const string& recipientState,
                  double weight, double costPerOunce, double flatFee)
        : Package(senderName, senderAddress, senderCity, senderState, senderZIP, recipientName, recipientAddress, recipientCity, recipien
                  weight, costPerOunce), flatFee(flatFee) {}

    double calculateCost() const override {
        return Package::calculateCost() + flatFee;
    }
};

class OvernightPackage : public Package {
private:
    double additionalFeePerOunce;

public:
    OvernightPackage(const string& senderName, const string& senderAddress, const string& senderCity, const string& senderState, const st
                     const string& recipientName, const string& recipientAddress, const string& recipientCity, const string& recipientSta
                     double weight, double costPerOunce, double additionalFeePerOunce)
        : Package(senderName, senderAddress, senderCity, senderState, senderZIP, recipientName, recipientAddress, recipientCity, recipien
                  weight, costPerOunce), additionalFeePerOunce(additionalFeePerOunce) {}

    double calculateCost() const override {
        return Package::calculateCost() + additionalFeePerOunce * weight;
    }
};

int main() {
    // Creating Package objects
    Package package1("John Doe", "123 Main St", "Anytown", "NY", "12345", "Jane Smith", "456 Elm St", "Othertown", "CA", "67890", 10, 0.5
    cout << "Standard Package Cost: $" << package1.calculateCost() << endl;

    // Creating TwoDayPackage objects
    TwoDayPackage twoDayPackage("John Doe", "123 Main St", "Anytown", "NY", "12345", "Jane Smith", "456 Elm St", "Othertown", "CA", "6789
    cout << "Two Day Package Cost: $" << twoDayPackage.calculateCost() << endl;

    // Creating OvernightPackage objects
    OvernightPackage overnightPackage("John Doe", "123 Main St", "Anytown", "NY", "12345", "Jane Smith", "456 Elm St", "Othertown", "CA",
    cout << "Overnight Package Cost: $" << overnightPackage.calculateCost() << endl;

    return 0;
}
```

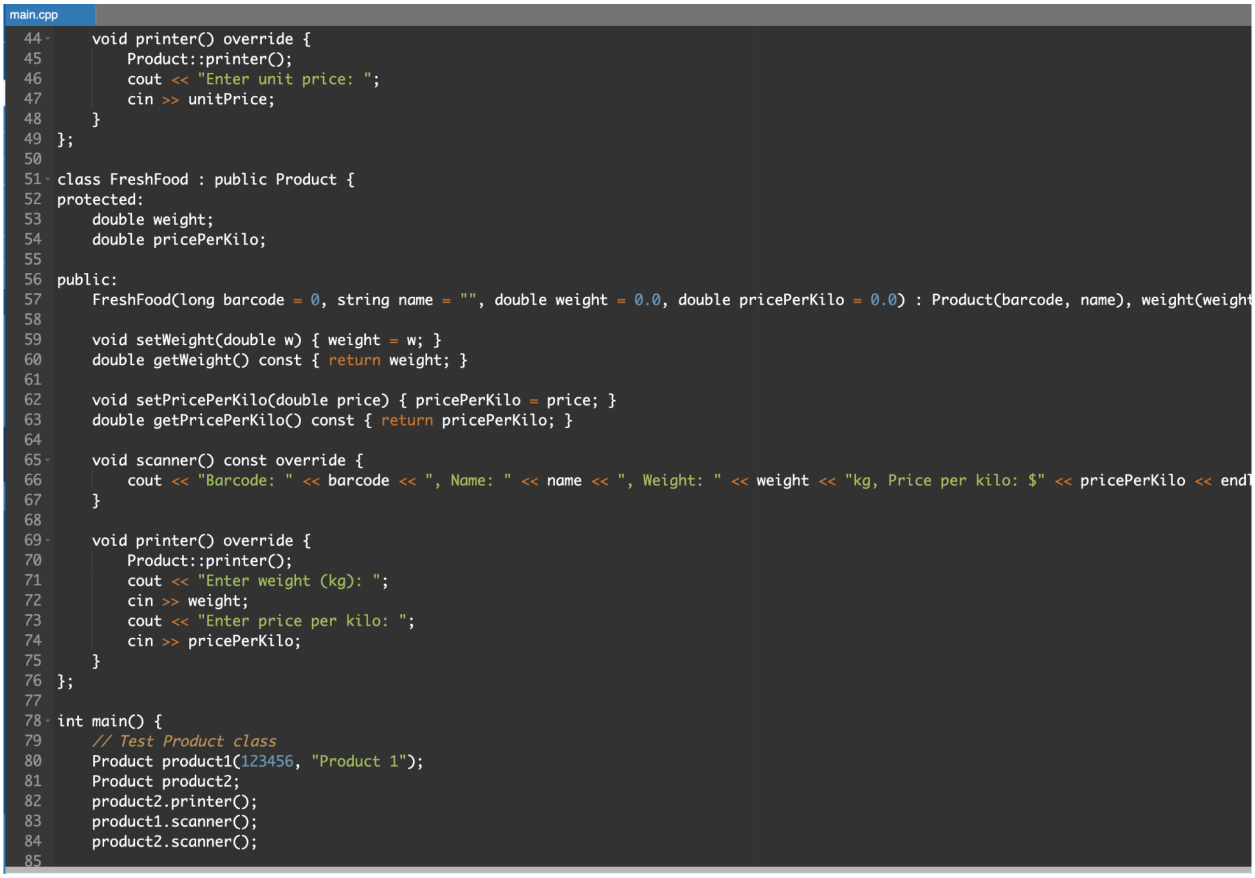


```
Standard Package Cost: $5
Two Day Package Cost: $10
Overnight Package Cost: $25
```

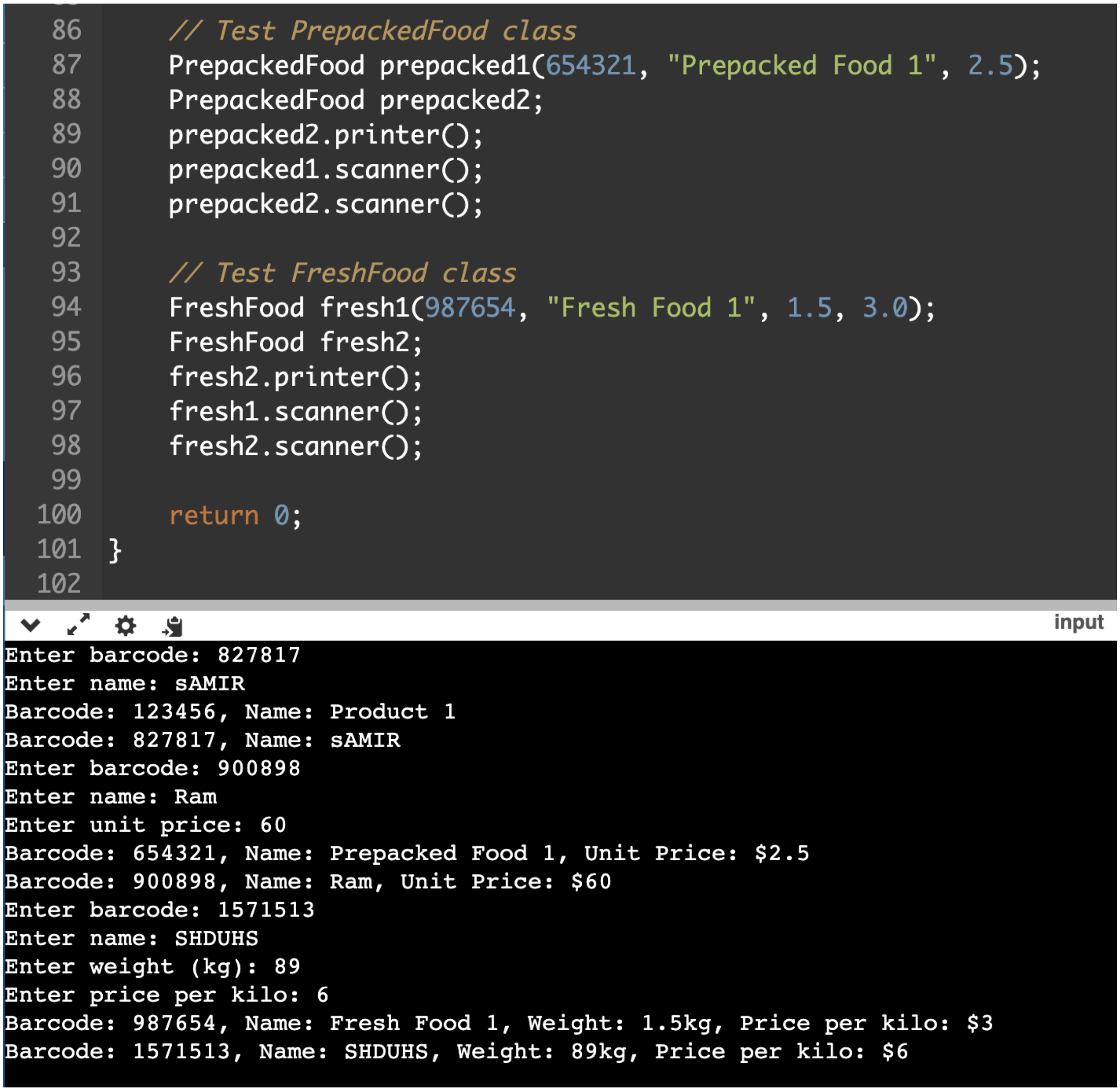
Question 2:



```cpp
#include <iostream>
#include <string>

using namespace std;

class Product {
protected:
    long barcode;
    string name;

public:
    Product(long barcode = 0, string name = "") : barcode(barcode), name(name) {}

    void setCode(long code) { barcode = code; }
    long getCode() const { return barcode; }

    virtual void scanner() const {
        cout << "Barcode: " << barcode << ", Name: " << name << endl;
    }

    virtual void printer() {
        cout << "Enter barcode: ";
        cin >> barcode;
        cout << "Enter name: ";
        cin.ignore();
        getline(cin, name);
    }
};

class PrepackedFood : public Product {
protected:
    double unitPrice;

public:
    PrepackedFood(long barcode = 0, string name = "", double unitPrice = 0.0) : Product(barcode, name), unitPrice(unitPrice) {}

    void setUnitPrice(double price) { unitPrice = price; }
    double getUnitPrice() const { return unitPrice; }

    void scanner() const override {
        cout << "Barcode: " << barcode << ", Name: " << name << ", Unit Price: $" << unitPrice << endl;
    }
```



```cpp
    void printer() override {
        Product::printer();
        cout << "Enter unit price: ";
        cin >> unitPrice;
    }
};

class FreshFood : public Product {
protected:
    double weight;
    double pricePerKilo;

public:
    FreshFood(long barcode = 0, string name = "", double weight = 0.0, double pricePerKilo = 0.0) : Product(barcode, name), weight(weight

    void setWeight(double w) { weight = w; }
    double getWeight() const { return weight; }

    void setPricePerKilo(double price) { pricePerKilo = price; }
    double getPricePerKilo() const { return pricePerKilo; }

    void scanner() const override {
        cout << "Barcode: " << barcode << ", Name: " << name << ", Weight: " << weight << "kg, Price per kilo: $" << pricePerKilo << endl
    }

    void printer() override {
        Product::printer();
        cout << "Enter weight (kg): ";
        cin >> weight;
        cout << "Enter price per kilo: ";
        cin >> pricePerKilo;
    }
};

int main() {
    // Test Product class
    Product product1(123456, "Product 1");
    Product product2;
    product2.printer();
    product1.scanner();
    product2.scanner();
```



```
    // Test PrepackedFood class
    PrepackedFood prepacked1(654321, "Prepacked Food 1", 2.5);
    PrepackedFood prepacked2;
    prepacked2.printer();
    prepacked1.scanner();
    prepacked2.scanner();

    // Test FreshFood class
    FreshFood fresh1(987654, "Fresh Food 1", 1.5, 3.0);
    FreshFood fresh2;
    fresh2.printer();
    fresh1.scanner();
    fresh2.scanner();

    return 0;
}

```

```
Enter barcode: 827817
Enter name: sAMIR
Barcode: 123456, Name: Product 1
Barcode: 827817, Name: sAMIR
Enter barcode: 900898
Enter name: Ram
Enter unit price: 60
Barcode: 654321, Name: Prepacked Food 1, Unit Price: $2.5
Barcode: 900898, Name: Ram, Unit Price: $60
Enter barcode: 1571513
Enter name: SHDUHS
Enter weight (kg): 89
Enter price per kilo: 6
Barcode: 987654, Name: Fresh Food 1, Weight: 1.5kg, Price per kilo: $3
Barcode: 1571513, Name: SHDUHS, Weight: 89kg, Price per kilo: $6
```