Khoi Duong

Prof. Yang

CS360

8/4/2022

HW#5

1.

Source code:

Package.h

```
#ifndef PACKAGE_H
#define PACKAGE_H
#include <iostream>
#include <iomanip>
using namespace std;

class Package{
    public:
    friend ostream& operator<<( ostream& output, const Package& p){
        output << "Package send from " << p.sender_name << " to " << p.recipient_name
<< endl;
        output << "From: " << p.sender_address << ", " << p.sender_city << ", " <<
p.sender_state << " " << p.sender_zip << endl;
        output << "To: " << p.recipient_address << ", " << p.recipient_city << ",
" << p.recipient_state << " " << p.recipient_zip << endl;
        output << "Type: unknown" << endl;
        output << "Weight: " << std::setprecision(4) << p.weight << endl;
        output << "Cost: " << std::setprecision(4) << p.cost << endl;
        return output;
    };
    string sender_name, sender_address, sender_city, sender_state;
    string recipient_name, recipient_address, recipient_city, recipient_state;
    int sender_zip, recipient_zip;
    double weight, cost_per_ounce, cost;
    Package(string a, string b, string c, string d, string e, string f, string g,
string h,
     int  i,  int  j,  double  k,  double  l):  sender_name(a),  sender_address(b),
sender_city(c), sender_state(d),
     recipient_name(e),          recipient_address(f),          recipient_city(g),
recipient_state(h), sender_zip(i), recipient_zip(j),
     weight(k > 0? k : throw invalid_argument( "Weight must be greater than 0" )),
     cost_per_ounce(l  >  0?  l  :  throw  invalid_argument(  "Cost  per  ounce  must  be
greater than 0" )){};
    ~Package(){};
    virtual double calculateCost(){
        cost = weight * cost_per_ounce;
        return cost;
    }
    virtual void getSender(){
        cout << "Sender name: " << sender_name << endl;
        cout << "Sender address: " << sender_address << endl;
        cout << "Sender city: " << sender_city << endl;
        cout << "Sender state: " << sender_state << endl;
        cout << "Sender zip: " << sender_zip << endl;
    }
    virtual void getRecipient(){
        cout << "Recipient name: " << recipient_name << endl;
        cout << "Recipient address: " << recipient_address << endl;
        cout << "Recipient city: " << recipient_city << endl;
        cout << "Recipient state: " << recipient_state << endl;
        cout << "Recipient zip: " << recipient_zip << endl;
    }
    virtual void getInfoPackage(){
        cout << "Weight: " << std::setprecision(4) << weight << endl;
        cout  <<  "Cost  per  ounce:  "  <<  std::setprecision(4)  <<  cost_per_ounce  <<
endl;
        cout << "Cost: " << std::setprecision(4) << cost << endl;
    }
};

#endif // PACKAGE_H
```

TwoDayPackage.h

```
#ifndef TWODAYPACKAGE_H
#define TWODAYPACKAGE_H
#include <iostream>
#include <iomanip>
#include "Package.h"
using namespace std;
```

```
class TwoDayPackage : public Package {
    public:
    friend ostream& operator<<( ostream& output, const TwoDayPackage& p){
        output << "Package send from " << p.sender_name << " to " << p.recipient_name
<< endl;
        output << "From: " << p.sender_address << ", " << p.sender_city << ", " <<
p.sender_state << " " << p.sender_zip << endl;
        output << "To: " << p.recipient_address << ", " << p.recipient_city << ",
" << p.recipient_state << " " << p.recipient_zip << endl;
        output << "Type: Two Day Package" << endl;
        output << "Weight: " << std::setprecision(4) << p.weight << endl;
        output << "Cost: " << std::setprecision(4) << p.cost << endl;
        return output;
    };
    double flat_fee;
    TwoDayPackage(string a, string b, string c, string d, string e, string f, string
g, string h,
     int i, int j, double k, double l, double fee): Package(a, b, c, d, e, f, g, h,
i, j, k, l),
     flat_fee(fee > 0? fee : throw invalid_argument("Flat fee must be greater than
zero")){};
    ~TwoDayPackage(){};
    double calculateCost(){
        cost = weight * cost_per_ounce + flat_fee;
        return cost;
    }
    void getInfoPackage(){
        cout << "Type: TwoDayPackage" << endl;
        cout << "Weight: " << weight << endl;
        cout << "Cost per ounce: " << std::setprecision(4) << cost_per_ounce <<
endl;
        cout << "Flat fee: " << std::setprecision(4) << flat_fee << endl;
        cout << "Cost: " << std::setprecision(4) << cost << endl;
    }
};

#endif // TWODAYPACKAGE_H
```

<u>OvernightPackage.h</u>

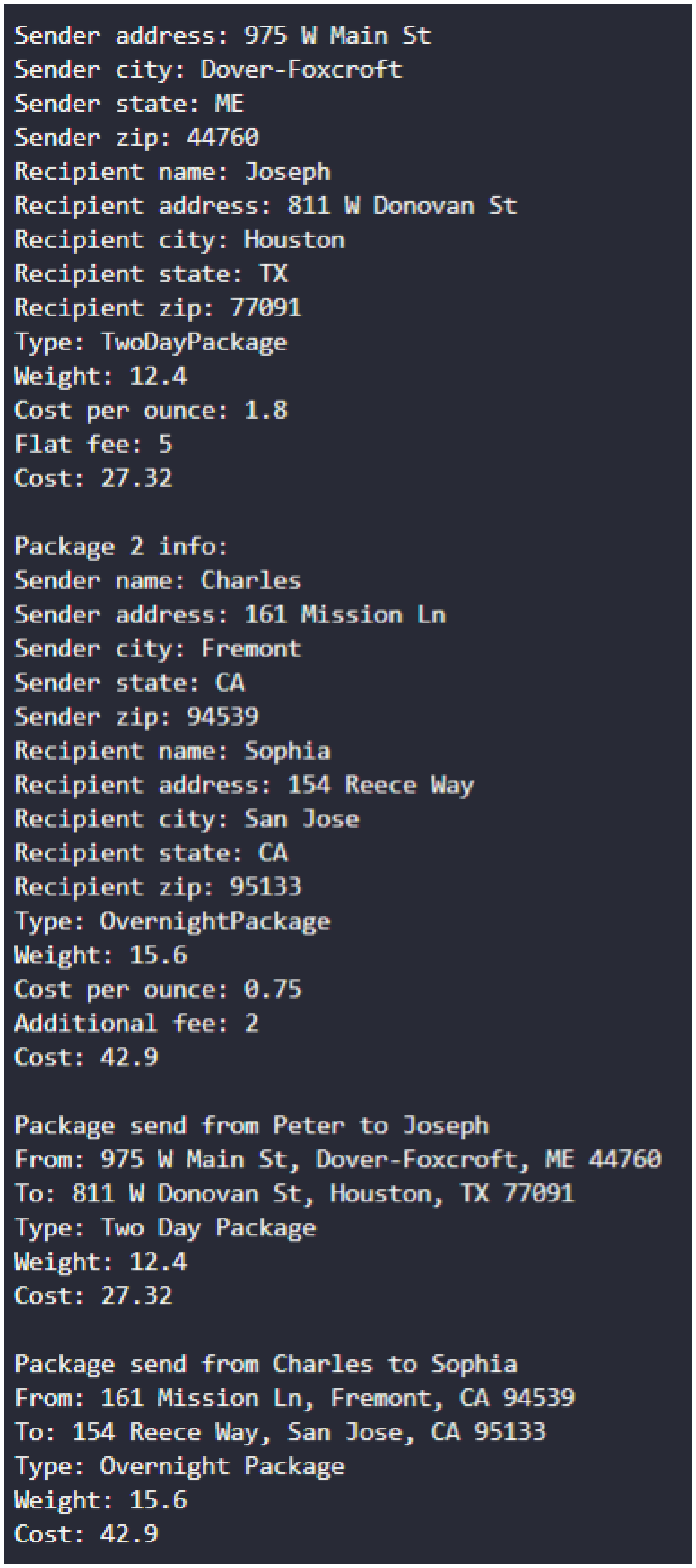
```
#ifndef OVERNIGHTPACKAGE_H
#define OVERNIGHTPACKAGE_H
#include <iostream>
```

```cpp
#include <iomanip>
#include "Package.h"
using namespace std;

class OvernightPackage : public Package {
    public:
    friend ostream& operator<<( ostream& output, const OvernightPackage& p){
        output << "Package send from " << p.sender_name << " to " << p.recipient_name << endl;
        output << "From: " << p.sender_address << ", " << p.sender_city << ", " << p.sender_state << " " << p.sender_zip << endl;
        output << "To: " << p.recipient_address << ", " << p.recipient_city << ", " << p.recipient_state << " " << p.recipient_zip << endl;
        output << "Type: Overnight Package" << endl;
        output << "Weight: " << std::setprecision(4) << p.weight << endl;
        output << "Cost: " << std::setprecision(4) << p.cost << endl;
        return output;
    };
    double additional_fee;
    OvernightPackage(string a, string b, string c, string d, string e, string f, string g, string h,
     int i, int j, double k, double l, double fee): Package(a, b, c, d, e, f, g, h, i, j, k, l),
     additional_fee(fee > 0? fee : throw invalid_argument("Additional fee must be greater than zero")){};
    ~OvernightPackage(){};
    double calculateCost(){
        cost = weight * (cost_per_ounce + additional_fee);
        return cost;
    }
    void getInfoPackage(){
        cout << "Type: OvernightPackage" << endl;
        cout << "Weight: " << weight << endl;
        cout << "Cost per ounce: " << std::setprecision(4) << cost_per_ounce << endl;
        cout << "Additional fee: " << std::setprecision(4) << additional_fee << endl;
        cout << "Cost: " << std::setprecision(4) << cost << endl;
    }
};

#endif // OVERNIGHTPACKAGE_H
```

<u>Main.cpp</u>

```cpp
#include <iostream>
using namespace std;
#include "Package.h"
#include "TwoDayPackage.h"
#include "OvernightPackage.h"

int main(){
    TwoDayPackage p1("Peter", "975 W Main St", "Dover-Foxcroft", "ME", "Joseph",
"811 W Donovan St", "Houston", "TX",
     44760, 77091, 12.4, 1.8, 5.0);
    cout << "Package 1 info: " << endl;
    p1.calculateCost();
    p1.getSender();
    p1.getRecipient();
    p1.getInfoPackage();
    cout << endl;
    OvernightPackage p2("Charles", "161 Mission Ln", "Fremont", "CA", "Sophia",
"154 Reece Way", "San Jose", "CA",
     94539, 95133, 15.6, 0.75, 2.0);
    cout << "Package 2 info: " << endl;
    p2.calculateCost();
    p2.getSender();
    p2.getRecipient();
    p2.getInfoPackage();
    cout << endl;

    cout << p1 << endl;
    cout << p2 << endl;
    return 0;
}
```

Run program & result:


```
Sender address: 975 W Main St
Sender city: Dover-Foxcroft
Sender state: ME
Sender zip: 44760
Recipient name: Joseph
Recipient address: 811 W Donovan St
Recipient city: Houston
Recipient state: TX
Recipient zip: 77091
Type: TwoDayPackage
Weight: 12.4
Cost per ounce: 1.8
Flat fee: 5
Cost: 27.32

Package 2 info:
Sender name: Charles
Sender address: 161 Mission Ln
Sender city: Fremont
Sender state: CA
Sender zip: 94539
Recipient name: Sophia
Recipient address: 154 Reece Way
Recipient city: San Jose
Recipient state: CA
Recipient zip: 95133
Type: OvernightPackage
Weight: 15.6
Cost per ounce: 0.75
Additional fee: 2
Cost: 42.9

Package send from Peter to Joseph
From: 975 W Main St, Dover-Foxcroft, ME 44760
To: 811 W Donovan St, Houston, TX 77091
Type: Two Day Package
Weight: 12.4
Cost: 27.32

Package send from Charles to Sophia
From: 161 Mission Ln, Fremont, CA 94539
To: 154 Reece Way, San Jose, CA 95133
Type: Overnight Package
Weight: 15.6
Cost: 42.9
```

2.

Source code:

<u>Product.h</u>

```
#ifndef PRODUCT_H
#define PRODUCT_H

#include <iostream>
using namespace std;

class Product{
    public:
    long barcode;
    string product_name;
    Product(long a, string b): barcode(a), product_name(b){}; // constructor
    Product(): barcode(0), product_name("Nothing"){}; // default constructor
    ~Product(){}; // destructor
    virtual void setCode(long a, string b){
        barcode = a;
        product_name = b;
    }
    virtual void getCode(){
        cout << "Barcode: " << barcode << endl;
        cout << "Product name: " << product_name << endl;
    }
    virtual void scanner(){
        cin >> barcode >> product_name;
        Product(barcode, product_name);
        cout << "Product name: " << product_name << ", barcode: " << barcode << "
is scanned." << endl;
    }
    virtual void printer(){
        cout << "Product name: " << product_name << ", barcode: " << barcode <<
endl;
    }
};

#endif // PRODUCT_H
```

<u>PrepackedFood.h</u>

```
#ifndef PREPACKEDFOOD_H
```

```
#define PREPACKEDFOOD_H
#include <iostream>
#include "Product.h"
using namespace std;

class PrepackedFood : public Product {
public:
    double unit_price;
    PrepackedFood(long a, string b, double c): Product(a, b), unit_price(c){}; //
constructor
    PrepackedFood(): Product(), unit_price(0){}; // default constructor
    ~PrepackedFood() {};
    void scanner(){
        cout << "Input barcode, product name, and unit price: ";
        cin >> barcode >> product_name >> unit_price;
        PrepackedFood(barcode, product_name, unit_price);
        cout << "Product name: " << product_name << ", barcode: " << barcode << ",
unit price: " << unit_price << " is scanned." << endl;
    };
    void printer(){
        cout << "Product name: " << product_name << ", barcode: " << barcode << ",
unit price: " << unit_price << endl;
    };
};

#endif // PREPACKEDFOOD_H
```

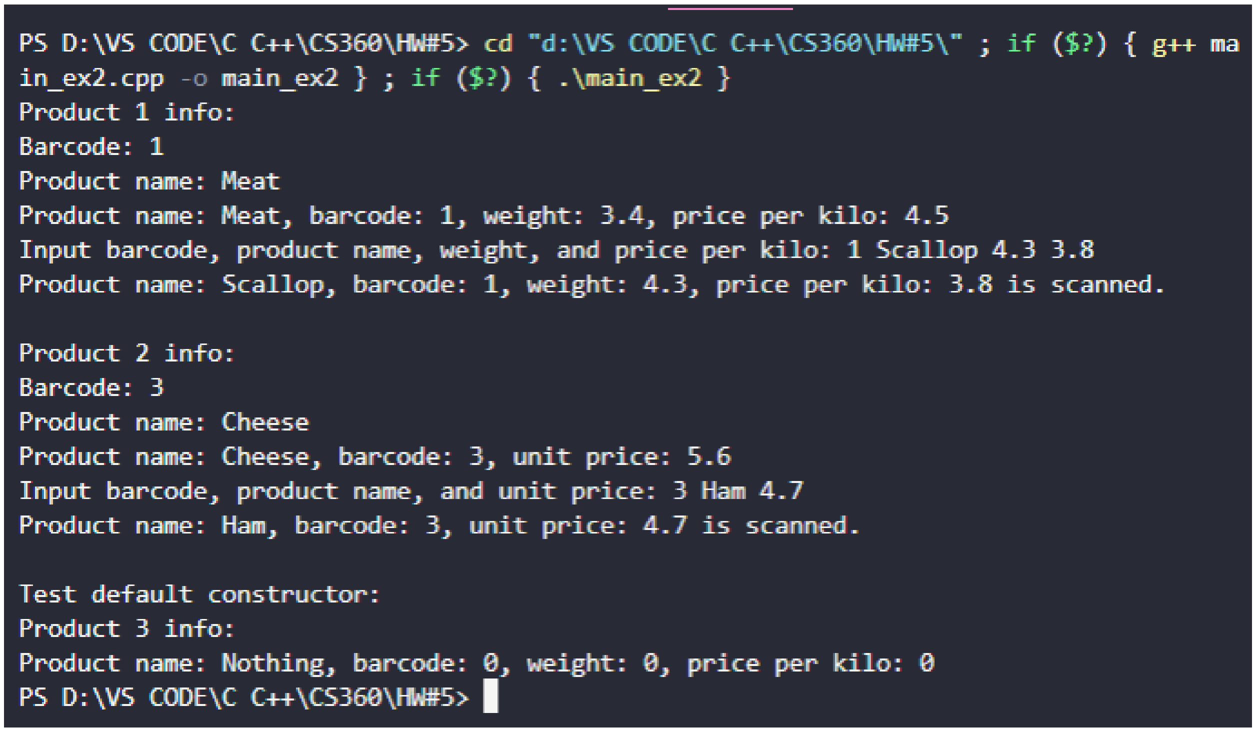
FreshFood.h

```
#ifndef FRESHFOOD_H
#define FRESHFOOD_H
#include <iostream>
#include "Product.h"
using namespace std;

class FreshFood : public Product {
public:
    double weight, price_per_kilo;
    FreshFood(long a, string b, double c, double d): Product(a, b), weight(c),
price_per_kilo(d){}; // constructor
    FreshFood(): Product(), weight(0), price_per_kilo(0){}; // default constructor
    ~FreshFood() {};
    void scanner(){
        cout << "Input barcode, product name, weight, and price per kilo: ";
        cin >> barcode >> product_name >> weight >> price_per_kilo;
        FreshFood(barcode, product_name, weight, price_per_kilo);
        cout << "Product name: " << product_name << ", barcode: " << barcode << ", weight: " << weight << ", price per kilo: " << price_per_kilo << " is scanned." << endl;
    };
    void printer(){
        cout << "Product name: " << product_name << ", barcode: " << barcode << ", weight: " << weight << ", price per kilo: " << price_per_kilo << endl;
    };
};

#endif // FRESHFOOD_H
```

<u>Main.cpp</u>

```
#include <iostream>
using namespace std;
#include "Product.h"
#include "FreshFood.h"
#include "PrepackedFood.h"

int main(){
    FreshFood p1(1, "Fish", 3.4, 4.5);
    cout << "Product 1 info: " << endl;
    p1.setCode(1, "Meat");
    p1.getCode();
    p1.printer();
    p1.scanner();
    cout << endl;
    PrepackedFood p2(3, "Ham", 5.6);
    cout << "Product 2 info: " << endl;
    p2.setCode(3, "Cheese");
    p2.getCode();
    p2.printer();
    p2.scanner();
    cout << endl;
    cout << "Test default constructor: " << endl;
    FreshFood p3;
    cout << "Product 3 info: " << endl;
    p3.printer();
    return 0;
}
```

Run program & result:



```
PS D:\VS CODE\C C++\CS360\HW#5> cd "d:\VS CODE\C C++\CS360\HW#5\" ; if ($?) { g++ ma
in_ex2.cpp -o main_ex2 } ; if ($?) { .\main_ex2 }
Product 1 info:
Barcode: 1
Product name: Meat
Product name: Meat, barcode: 1, weight: 3.4, price per kilo: 4.5
Input barcode, product name, weight, and price per kilo: 1 Scallop 4.3 3.8
Product name: Scallop, barcode: 1, weight: 4.3, price per kilo: 3.8 is scanned.

Product 2 info:
Barcode: 3
Product name: Cheese
Product name: Cheese, barcode: 3, unit price: 5.6
Input barcode, product name, and unit price: 3 Ham 4.7
Product name: Ham, barcode: 3, unit price: 4.7 is scanned.

Test default constructor:
Product 3 info:
Product name: Nothing, barcode: 0, weight: 0, price per kilo: 0
PS D:\VS CODE\C C++\CS360\HW#5>
```