#ifndef MERCHANT\_H

#define MERCHANT\_H

#include<iostream>

#include <string>

using namespace std;

class Merchant {

public:

Merchant();

Merchant(Antique antiques[], int quantities[]);

Merchant(float rev);

Merchant(const Merchant& origClass); // Copy constructor

~Merchant();

void haggle();

void printMenu();

void selectAntique(float& budget);

void leave(float budget);

Merchant operator= (Merchant& merc);

bool operator== (Merchant& other);

void addAntique(Antique ant, int quantity);

private:

Antique antAnt[10]; // NoName indicates name was not set

int\* quantities[10];

float revenue;

};

#endif

Merchant seller(antArr, stockArr);

merc= new Merchant[guildsize];

Merchant\* merc;

Merchant MerchantGuild::getMembers() {

Merchant\* array{ new Merchant[5]{ 9, 7, 5, 3, 1 } };

return \*array;

}

antAnt[i].setName(antAnt[i].getName());

antAnt[i].setPrice(antAnt[i].getPrice());

Merchant MerchantGuild::addMember() {

Merchant\* array{ new Merchant[5]{ 9, 7, 5, 3, 1 } };

return \*array;

}

Merchant addMember();

36-37=cout << "Error! File not found." << endl;

return 1;

cout << "KEY: True is: 1 False is: 0" << endl << endl;

cout << "antique test" << endl;

cout << "fork and knife: $5.75" << endl;

cout << "0 : Ans=0" << endl;

cout << "1 : Ans=1" << endl;

Merchant m1(1.2), m2(2.5);

cout << "merchant test" << endl;

cout << "0 : Ans=0" << endl;

cout << "1 : Ans=1" << endl;

Merchant m3(m1);

cout << "1 : Ans=1" << endl;

cout << "merchant guild tests" << endl;

MerchantGuild mg1;

mg1.addMember(m1);

Merchant\* tmp = mg1.getMembers();

cout << "1 : Ans=1" << endl;

cout << "1 : Ans=1" << endl;

return 1;

(merc.revenue == revenue) &&

(merc.antsize == antsize)&&

(count== antsize);

bool Merchant::operator==(const Merchant& merc) {

int count = 0;

for (int i = 0; i < antsize; i++) {

if((merc.antAnt[i].getName()== antAnt[i].getName())&&(merc.antAnt[i].getPrice()== antAnt[i].getPrice())&&(merc.quantities[i] = quantities[i])){

count++;

}

}

return (merc.revenue == revenue) &&

(merc.antsize == antsize)&&

(count== antsize);

}

void Merchant::addAntique(Antique ant, int quantity) {

Antique\* my\_arr = new Antique[antsize++];

int\* my\_quan = new int[antsize];

for (int i = 0; i < antsize-1; i++) {

my\_arr[i] = antAnt[i];

my\_quan[i] = quantities[i];

}

my\_arr[antsize].setName(ant.getName());

my\_arr[antsize].setPrice(ant.getPrice());

my\_quan[antsize]=quantity;

delete[] antAnt;

delete[] quantities;

antAnt = my\_arr;

quantities = my\_quan;

}

Antique\* antAnt = new Antique[10];

int\* quantities = new int[10];

antsize = 10;

revenue = 0;

for (int i = 0; i < antsize; i++) {

antAnt[i].setName(nullptr);

antAnt[i].setPrice(0);

quantities[i] = 10;

}

Merchant::Merchant(const Merchant& origClass) {

Antique\* antAnt = new Antique[antsize];

int\* quantities = new int[antsize];

\*antAnt = \*(origClass.antAnt);

for (int i = 0; i < antsize; i++) {

antAnt[i].setName(origClass.antAnt[i].getName());

antAnt[i].setPrice(origClass.antAnt[i].getPrice());

\*antAnt = \*(origClass.antAnt);

quantities[i] = origClass.quantities[i];

}

}

antAnt = new Antique[origClass.antsize];

quantities = new int[origClass.antsize];

for (int i = 0; i < origClass.antsize; i++) {

antAnt[i] = origClass.antAnt[i];

quantities[i] = origClass.quantities[i];

}

antsize = origClass.antsize;

revenue = origClass.revenue;

Merchant::Merchant(const Merchant& origClass) {

antsize = origClass.antsize;

revenue = origClass.revenue;

antAnt = new Antique[origClass.antsize];

quantities = new int[origClass.antsize];

for (int i = 0; i < origClass.antsize; i++) {

antAnt[i]=origClass.antAnt[i];

quantities[i] = origClass.quantities[i];

}

}

cout << bool(tmp[0]==m1) << " : Ans=1" << endl;

cout << bool((tmp[0] == m1)&& (tmp[1]==m2)) << " : Ans=1" << endl;

Antique a1;

Antique a2;

a2.setName("Painting");

Antique a3;

cout << "KEY: True is: 1 False is: 0" << endl << endl;

cout << "antique test" << endl;

cout << "fork and knife: $5.75" << endl;

cout << bool(a1 == a2)<<" : Ans=0" << endl;

cout << bool(a1 == a3) << " : Ans=1" << endl;

Merchant m1(1.2), m2(2.5);

cout << "merchant test" << endl;

cout << bool(m1 == m2) << " : Ans=0" << endl;

cout << bool(m1 == m1) << " : Ans=1" << endl;

Merchant m3(m1);

cout << bool(m3 == m1) << " : Ans=1" << endl;

cout << "merchant guild tests" << endl;

MerchantGuild mg1;

mg1.addMember(m1);

Merchant\* tmp = mg1.getMembers();

cout << "1 : Ans=1" << endl;

cout << "1 : Ans=1" << endl;

return 1;