// ========================

// Attached: HW\_3c, Invoices.h

// ========================

// Program: HW\_3c.ccp

// ========================

// Programmer: Milo Fisher

// Class: CS 1B

// ========================

#include<iostream>

#include<iomanip>

#include"Invoices.h"

using namespace std;

void setInvoice(Invoice&);

void displayInvoice(Invoice&);

int main()

{

Invoice inv\_1;

Invoice inv\_2;

setInvoice(inv\_1);

setInvoice(inv\_2);

system("CLS");

displayInvoice(inv\_1);

displayInvoice(inv\_2);

system("pause");

return 0;

}

void setInvoice(Invoice& inv)

{

int num;

double amt;

char paid;

cout << "Enter the invoice:\nINVOICE #: ";

cin >> num;

inv.setNum(num);

cout << "AMOUNT: ";

cin >> amt;

inv.setAmt(amt);

cout << "PAID (y/n): ";

cin >> paid;

if (toupper(paid) == 'Y')

inv.setPaid(true);

else

inv.setPaid(false);

cout << endl;

}

void displayInvoice(Invoice& inv)

{

cout << "---- Invoice Summary ----\nINVOICE #:" << setw(6) << inv.getNum() << "\nAMOUNT:" << setw(11) << fixed << setprecision(2) << inv.getAmt() << "\nPAID:";

if (inv.getPaid() == true)

cout << setw(13) << "Paid\n\n";

else

cout << setw(17) << "Not Paid\n\n";

}

// ======= OUTPUT =========

/\*

Enter the invoice:

INVOICE #: 1000

AMOUNT: 220.33

PAID (y/n): y

Enter the invoice:

INVOICE #: 1001

AMOUNT: 123.75

PAID (y/n): n

//screen clear

---- Invoice Summary ----

INVOICE #: 1000

AMOUNT: 220.33

PAID: Paid

---- Invoice Summary ----

INVOICE #: 1001

AMOUNT: 123.75

PAID: Not Paid

Press any key to continue . . .

\*/

// ========================

// ========================

// Attached: HW\_3c, Invoices.h

// ========================

// Program: Invoices.h

// ========================

// Programmer: Milo Fisher

// Class: CS 1A

// ========================

#pragma once

#include<iostream>

#include<string>

using namespace std;

struct Invoice

{

private:

int invoiceNum;

double amt;

bool paid;

public:

Invoice();

Invoice(int invoiceNum, double amt, bool paid);

void setNum(int);

void setAmt(double);

void setPaid(bool);

int getNum();

double getAmt();

bool getPaid();

~Invoice();

};

Invoice::Invoice()

{

invoiceNum = 0;

amt = 0.0;

paid = false;

}

Invoice::Invoice(int invoiceNum, double amt, bool paid)

{

this->invoiceNum = invoiceNum;

this->amt = amt;

this->paid = paid;

}

void Invoice::setNum(int invoiceNum)

{

this->invoiceNum = invoiceNum;

}

void Invoice::setAmt(double amt)

{

this->amt = amt;

}

void Invoice::setPaid(bool paid)

{

this->paid = paid;

}

int Invoice::getNum()

{

return invoiceNum;

}

double Invoice::getAmt()

{

return amt;

}

bool Invoice::getPaid()

{

return paid;

}

Invoice::~Invoice() {}