// comsc - 200

// main.cpp

// 7a

// completed

// Created by Jeff on 9/26/16.

// Copyright © 2016 Jeff zhang. All rights reserved.

//

#include <iostream>

#include <iomanip>

#include "ProductionWorker.h"

using namespace std;

int main() {

Employee e1("James Baker", "1501", "1/1/2015");

ProductionWorker pw1("John Jones", "1502", "1/1/2015", 2, 18.00);

ProductionWorker pw2("Jerry Jones", "1503", "1/2/2015", 1, 15.00);

cout << "\n---e1---";

cout << "\n...output e1 with .tostring()"

<< e1.toString()

<< "\n\n...output e1 with <<" << e1

<< "\n\n...output e1 with display(): ";

e1.display();

// repeat for pw1 and pw2

cout << "\n---pw1---";

cout << "\n...output e1 with .tostring()"

<< pw1.toString()

<< "\n\n...output pw1 with <<" << pw1

<< "\n\n...output pw1 with display(): ";

pw1.display();

cout << "\n---pw2---";

cout << "\n...output pw2 with .tostring()"

<< pw2.toString()

<< "\n\n...output pw2 with <<" << pw2

<< "\n\n...output pw2 with display(): ";

pw2.display();

}

//

// Employee.h

// 7a

//

// Created by Jeff on 9/26/16.

// Copyright © 2016 Jeff zhang. All rights reserved.

//

#ifndef Employee\_h

#define Employee\_h

#include <iostream>

#include <sstream>

#include <string>

#include "ProductionWorker.h"

using namespace std;

class Employee;

ostream &operator << (ostream&, const Employee &);

class Employee{

private:

string name;

string number;

string date;

public:

Employee(){

name="";number="";date="";

}

Employee(string nam, string num,string d){

name = nam;

number = num;

date = d;

}

void setName(string n){

name = n;

}

void setNumber(string n){

number =n;

}

void setDate(string d){

date = d;

}

string getName()const{

return name;

}

string getNumber()const{

return number;

}

string getDate()const{

return date;

}

void display();

string toString();

// pass by reference

friend ostream &operator << (ostream &, const Employee &);

};

void Employee::display(){

cout << "\n name" << this->getName() << "\n number : " << getName()

<< "\n date " << this->getDate() << endl;

}

string Employee::toString(){

stringstream ss;

ss<< "\n name" << this->getName() << "\n number : " << getName()

<< "\n date " << this->getDate() <<"this tostring in employee.h"<< endl;

return ss.str();

}

ostream &operator << (ostream &strm, const Employee &e){

strm<< "\n name" << e.getName() << "\n number : " << e.getName()

<< "\n date " << e.getDate() << endl;

return strm;

}

#endif /\* Employee\_h \*/

//

// ProductionWorker.h

// 7a

//

// Created by Jeff on 9/26/16.

// Copyright © 2016 Jeff zhang. All rights reserved.

//

#ifndef ProductionWorker\_h

#define ProductionWorker\_h

#include <string>

#include <sstream>

#include <iostream>

#include "Employee.h"

using namespace std;

class ProductionWorker : public Employee{

private:

int shift;

double rate;

public:

ProductionWorker() : Employee(){

shift = 0;

rate = 0.0;

}

ProductionWorker(string nam, string num,string d,int s,double r)

:Employee(nam,num,d){

shift = s;

rate =r;

}

void setShift(int s){

shift = s;

}

void setRate(double r){

rate = r;

}

int getShift()const{

return shift;

}

double getRate()const{

return rate;

}

string getShiftName()const{

if(shift ==1)

return "day";

else if (shift ==2)

return "Night";

return "ivalid";

}

string toString(){

string s = Employee::toString();

stringstream ss;

ss<< "\n shift" << this->getShiftName()

<<"\n Rate " << this->getRate() << " this to string in production"<< endl;

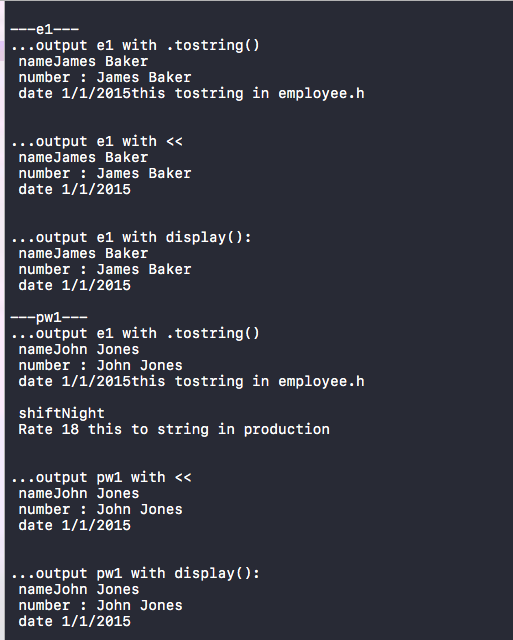
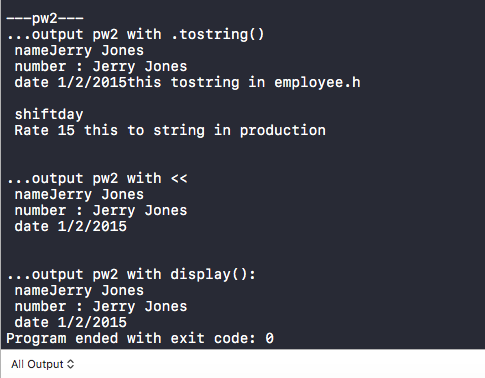
s += ss.str();

return s;

}

};

#endif /\* ProductionWorker\_h \*/



// comsc 200

// completed

// main.cpp

// 7c

// boli zhang

// Created by Jeff on 9/28/16.

// Copyright © 2016 Jeff zhang. All rights reserved.

//

#include <iostream>

#include "Date.h"

#include "Time.h"

class DateTime;

using namespace std;

ostream &operator<<(ostream&strm,const DateTime &d);

istream &operator<<(istream&strm, DateTime &d);

class DateTime : public Date,public Time{

public:

DateTime():Date(),Time(){};

DateTime(int m,int d,int y,int h, int mn, int s)

:Date(m,d,y),Time(h,mn,s){};

friend ostream &operator<<(ostream&strm,const DateTime &d);

friend istream &operator<<(istream&strm, DateTime &d);

};

ostream &operator<<(ostream&strm,const DateTime &dt){

strm << dt.getMonth()<<"/" << dt.getDay() << "/" << dt.getYear()<<" "<< dt.getHour()<<":" << dt.getMin() << ":" << dt.getSecond();

return strm;

};

istream &operator>>(istream&strm, DateTime &dt){

int temp;

cout << " \n Enter the month: \n\t ";

strm >> temp; dt.setMonth(temp);

cout << " \n Enter the day: \n\t ";

strm >> temp; dt.setDay(temp);

cout << " \n Enter the year: ";

strm >> temp; dt.setYear(temp);

cout << " \n Enter the hour: ";

strm >> temp; dt.setHour(temp);

cout << " \n Enter the min: ";

strm >> temp; dt.setMin(temp);

cout << " \n Enter the second: ";

strm >> temp; dt.setSecond(temp);

return strm;

};

int main(){

Date oneDay;

cout << "\nDate oneDate; " << oneDay;

Time oneTime;

cout << "\nTime oneTime; " << oneTime;

DateTime oneDayTime;

cout << "\nDateTime oneDayTime; " << oneDayTime;

Date toDay(9, 30, 2015);

cout << "\nDate toDay(9, 30, 2015); " << toDay;

Time toTime(8, 11, 22);

cout << "\nTime toTime(8, 11, 22); " << toTime;

DateTime toDayTime(9, 30, 2015, 8, 11, 22);

cout << "\nDateTime toDayTime(9, 30, 2015, 8, 11, 22); \n" << toDayTime << endl;

cout << "Enter a Date: ";

cin >> oneDay;

cout << "\n ... You have entered: " << oneDay;

cout << "\n Enter a Time: ";

cin >> oneTime;

cout << "\n ... You have entered: " << oneTime;

cout << "\n Etner a DateTime: ";

cin >> oneDayTime;

cout << "\n ... You have entered: " << oneDayTime <<endl;

}

// comsc 200

// Date.h

// 7c

// boli zhang

// completed

// Created by Jeff on 9/28/16.

// Copyright © 2016 Jeff zhang. All rights reserved.

//

#ifndef Date\_h

#define Date\_h

#include <iostream>

using namespace std;

class DateTime;

ostream &operator<<(ostream&,const DateTime &);

istream &operator>>(istream&, DateTime &);

class Date{

private:

int month;

int day;

int year;

public:

Date():month(1),day(1),year(2016){};

Date(int m,int d,int y):month(m),day(d),year(y){};

int getMonth()const {return month;};

int getDay()const {return day;};

int getYear()const{return year;};

void setMonth(int m) {

month =m;

};

void setDay(int d) {

day =d;

};

void setYear(int y) {

year = y;

};

friend ostream &operator <<(ostream &strm,const Date &);

friend istream &operator>>(istream &strm, Date&);

};

ostream &operator <<(ostream &strm,const Date &d){

strm << d.getMonth()<<"/" << d.getDay() << "/" << d.getYear()<<" ";

return strm;

};

istream &operator>>(istream&strm, Date&d){

int temp;

cout << " \n Enter the month: \n\t ";

strm >> temp; d.setMonth(temp);

cout << " \n Enter the month: \n\t ";

strm >> temp; d.setDay(temp);

cout << " \n Enter the month: \n\t ";

strm >> temp; d.setYear(temp);

return strm;

};

#endif /\* Date\_h \*/

// comsc 200

// Time.h

// 7c

// boli zhang

// completed

// Created by Jeff on 9/28/16.

// Copyright © 2016 Jeff zhang. All rights reserved.

//

#ifndef Time\_h

#define Time\_h

#include <iostream>

using namespace std;

class Time;

ostream &operator <<(ostream&,const Time &d);

istream &operator>>(istream&, Time&d);

class Time {

private:

int hour;

int min;

int sec;

public:

Time():hour(12),min(30),sec(23){};

Time(int h,int m,int s) : hour(h),min(m),sec(s){};

int getHour()const {return hour;};

int getMin()const {return min;};

int getSecond()const{return sec;};

void setHour(int h) {

hour =h;

};

void setMin(int m) {

min =m;

};

void setSecond(int s) {

sec = s;

};

friend ostream &operator <<(ostream &strm,const Time &d);

friend istream &operator >>(istream &strm, Date &d);

};

ostream &operator <<(ostream&strm,const Time &d){

strm << d.getHour()<<":" << d.getMin() << ":" << d.getSecond();

return strm;

};

istream &operator>>(istream&strm, Time&d){

int temp;

cout << " \n Enter the hour: ";

strm >> temp; d.setHour(temp);

cout << " \n Enter the min: ";

strm >> temp; d.setMin(temp);

cout << " \n Enter the second: ";

strm >> temp; d.setSecond(temp);

return strm;

};

#endif /\* Time\_h \*/

