//

// main.cpp

//comsc 200

//compeleted

//boli zhang

//As4.1

//

// Created by Jeff on 9/10/16.

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

//

#include <iostream>

#include <vector>

#include <string>

#include "PayRoll.h"

#include <sstream>

using namespace std;

int main() {

PayRoll data;

double rate;

int size,hours;

string input;

std::vector<double> myvector;

while(true){

cout<< "Enter the number of employees: ";

getline(cin,input);

stringstream ss(input);

ss >> size;

if(!ss.fail())break;

cout << input<<" is not a number. \n";

}

cout<< "Enter the hours worked and pay rate for "<<size << " employees: \n";

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

cout << "Employee # " <<i+1 << " Pay rate: ";

while (true) {

getline(cin,input);

stringstream ss(input);

ss >> rate; // floating number

if(!ss.fail())break;

cout << input<<" is not a valid rate. \n"<<"Employee # " << i+1 << " Pay rate: ";

;

data.setRate(rate);

}

cout << "Employee # " <<i+1 << " hours worked: ";

while (true) {

getline(cin,input);

stringstream ss(input);

ss >> hours; //int type

if(hours>60){

//cout <<"invalid number of hours. \n" ;

}

else

if(!ss.fail())break;

cout << input<<" is not a valid rate."<< endl <<"Employee # " << i+1 << " hours worked: ";

data.setRate(hours);

}

//std::vector<double> myvector;

double tt = rate \* hours;

myvector.push\_back(tt);

}

std::cout << "Total pay : \n";

for(int i=1;i<size;i++)

for (std::vector<double>::iterator it = myvector.begin() ; it != myvector.end(); ++it)

std::cout << " Employee # " <<i++<< ": "<<' ' << \*it << "\n";

std::cout << '\n';

}

//comsc 200

//compeleted

//boli zhang

// PayRoll.h

// As4.1

//

// Created by Jeff on 9/10/16.

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

//

#ifndef PayRoll\_h

#define PayRoll\_h

class PayRoll{

private:

double rate;

int hours;

public:

//constructor

PayRoll(){

rate = 0; hours = 0;

}

PayRoll(double r,int h){

setRate(r);

setHours(h);

}

//setter

void setRate(double r){

rate = r;

}

void setHours(int h){

hours = h;

}

//getter

int getHours(){

return hours;

}

double getRate(){

return rate;

};

//feature

void totall(){

std::vector<double> myvector;

double tt = rate \* hours;

std::cout <<"rate: " << getRate()<< "\n";

myvector.push\_back(tt);

std::cout << "myvector contains:";

for (std::vector<double>::iterator it = myvector.begin() ; it != myvector.end(); ++it)

std::cout << ' ' << \*it;

std::cout << '\n';

}

};

#endif /\* PayRoll\_h \*/

