Michelle Zulli

Dr. Alireza Ebrahimi

Introduction to C++ and OOP

August 2, 2015

# Module 5: Payroll

The program was modified to use a class called employee. Data members and functions were placed in the class. A standard tax rate of .3 was applied for all employees.

## Code

#include<iostream>

#include<fstream>

#include<iomanip>

using namespace std;

class employee {

ifstream fin;

public: string firstname[6], lastname[6];

int id[6];

float hours[6], base\_hours[6], ot\_hours[6], rate[6], ot\_rate[6], grosspay[6], netpay[6];

float taxamount[6], taxrate;

char status[6];

employee();

~employee();

int readdata(string[], string[], char[], int[], float[], float[], int);

void findhours(float[], float[], float[], int);

void findotrate(float[], float[], int);

void findgrosspay(float[], float[], float[], float[], float[], int);

void findtaxamt(float[], float[], float, int);

void findnetpay(float[], float[], float[], int);

float findavgnet(float[], int);

void printdata(string[], string[], char[], int[], float[], float[], float[], float[], float[], float, float[], float, int);

};

// function definitions

employee::employee() {

fin.open("employees.in");

taxrate = .3;

}

employee::~employee() {

fin.close();

}

// pull data from the input file to the array

int employee::readdata(string firstname[], string lastname[], char status[], int id[], float hours[], float rate[], int n) {

n = 0;

while (fin >> firstname[n] >> lastname[n] >> status[n] >> id[n] >> hours[n] >> rate[n]) n++;

return n;

} // readdata

// set base and ot hours

void employee::findhours(float base\_hours[], float hours[], float ot\_hours[], int n){

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

base\_hours[i] = hours[i];

ot\_hours[i] = 0;

if (base\_hours[i] > 40) {

ot\_hours[i] = base\_hours[i] - 40;

base\_hours[i] = 40;

}

}

} // findhours

// set ot rate

void employee::findotrate(float ot\_rate[], float rate[], int n){

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

ot\_rate[i] = rate[i] \* 1.5;

}

} // findotrate

// calculate gross pay

void employee::findgrosspay(float grosspay[], float base\_hours[], float rate[], float ot\_hours[], float ot\_rate[], int n){

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

grosspay[i] = (base\_hours[i] \* rate[i]) + (ot\_hours[i] \* ot\_rate[i]);

}

} // findgrosspay

// calculate tax amount

void employee::findtaxamt(float taxamount[], float grosspay[], float taxrate, int n){

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

taxamount[i] = grosspay[i] \* taxrate;

}

} // findtaxamt

// calculate net pay

void employee::findnetpay(float netpay[], float grosspay[], float taxamount[], int n){

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

netpay[i] = grosspay[i] - taxamount[i];

}

} // findnetpay

// calculate average net pay

float employee::findavgnet(float netpay[], int n){

float sum = 0;

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

sum += netpay[i];

}

return sum / n;

}

// print data table

void employee::printdata(string firstname[], string lastname[], char status[], int id[], float hours[], float ot\_hours[], float rate[], float ot\_rate[], float grosspay[], float taxrate, float netpay[], float avg, int n){

// print table header

cout << setiosflags(ios::left)

<< " ZULLI PAYROLL"

<< endl

<< setw(16) << "FIRST NAME" << setw(16) << "LAST NAME"

<< setw(8) << "STATUS" << setw(6) << "ID" << setw(8) << "HOURS"

<< setw(10) << "OT HOURS" << setw(10) << "RATE"

<< setw(10) << "OT RATE" << setw(10) << "GROSS"

<< setw(10) << "TAX" << setw(10) << "NET"

<< endl

<< "=============== =============== ======= ===== ======= "

<< "========= ========= ========= ========= ========= ========="

<< endl;

// print employee data

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

cout << setprecision(2)

<< setiosflags(ios::showpoint | ios::fixed | ios::left)

<< endl

<< setw(16) << firstname[i] << setw(16) << lastname[i]

<< setw(8) << status[i] << setw(6) << id[i]

<< setw(8) << hours[i] << setw(10) << ot\_hours[i] << setw(10) << rate[i]

<< setw(10) << ot\_rate[i] << setw(10) << grosspay[i]

<< setw(10) << taxrate << setw(10) << netpay[i]

<< endl;

}

// print net pay info

cout << endl

<< "AVERAGE NET PAY FOR ALL EMPLOYEES: " << avg

<< endl;

} // printdata

int main() {

employee employee;

int n = employee.readdata(employee.firstname, employee.lastname, employee.status, employee.id, employee.hours, employee.rate, 6);

employee.findhours(employee.base\_hours, employee.hours, employee.ot\_hours, n);

employee.findotrate(employee.ot\_rate, employee.rate, n);

employee.findgrosspay(employee.grosspay, employee.base\_hours, employee.rate, employee.ot\_hours, employee.ot\_rate, n);

employee.findtaxamt(employee.taxamount, employee.grosspay, employee.taxrate, n);

employee.findnetpay(employee.netpay, employee.grosspay, employee.taxamount, n);

float avg = employee.findavgnet(employee.netpay, 6);

employee.printdata(employee.firstname, employee.lastname, employee.status, employee.id, employee.hours, employee.ot\_hours, employee.rate, employee.ot\_rate, employee.grosspay, employee.taxrate, employee.netpay, avg, n);

return 0;

} // MAIN

## Output

