Michelle Zulli

Dr. Alireza Ebrahimi

Introduction to C++ and OOP

June 28, 2015

# Module 3: Payroll

## A: Output Formatted in a Table — Code

#include<iostream>

#include<fstream>

#include<iomanip>

using namespace std;

int main() {

string firstname, lastname;

int id;

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

float taxamount, taxrate;

char status;

// 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;

// loop through input file

ifstream fin("employee5A.in");

while(fin >> firstname >> lastname >> status >> id >> hours >> rate){

// set base and ot hours

if(hours <= 40) {

base\_hours = hours;

ot\_hours = 0;

}

else {

base\_hours = 40;

ot\_hours = hours - 40;

}

// calculate gross pay

ot\_rate = rate \* 1.5;

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

// calculate tax rate from gross

if(grosspay > 1000){

taxrate = .3;

}

else if(grosspay > 800){

taxrate = .2;

}

else if(grosspay > 500){

taxrate = .1;

}

else{

taxrate = 0;

}

// adjust tax rate based on status

if(status == 's' || status == 'S'){

taxrate += .05;

}

else if(status == 'h' || status == 'H'){

taxrate -= .05;

}

if(taxrate < 0) {

taxrate = 0;

}

// calculate tax and net pay

taxamount = grosspay \* taxrate;

netpay = grosspay - taxamount;

// print employee data

cout << setprecision(2)

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

<< endl

<< setw(16) << firstname << setw(16) << lastname

<< setw(8) << status << setw(6) << id

<< setw(8) << hours << setw(10) << ot\_hours << setw(10) << rate

<< setw(10) << ot\_rate << setw(10) << grosspay

<< setw(10) << taxrate << setw(10) << netpay

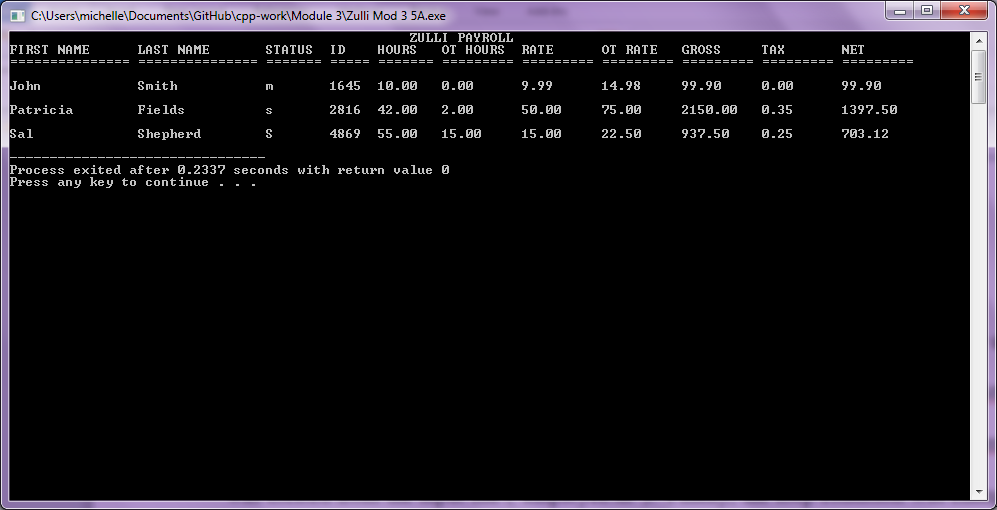
<< endl;

}//WHILE

return 0;

}//MAIN

## A: Output Formatted in a Table — Output



## B: Arrays and Loops — Code

#include<iostream>

#include<fstream>

#include<iomanip>

using namespace std;

int main() {

const int SIZE = 3; // set array size

// declare arrays

string firstname[SIZE], lastname[SIZE];

int id[SIZE];

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

float taxamount[SIZE], taxrate[SIZE];

char status[SIZE];

// pull data from the input file to the array

ifstream fin("employee5A.in");

int n = 0;

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

// compute overtime pay

int i = 0;

while (i < n) {

// set base and ot hours

if(hours[i] <= 40) {

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

ot\_hours[i] = 0;

}

else {

base\_hours[i] = 40;

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

}

// set overtime pay

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

// calculate gross pay

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

// calculate tax rate from gross

if(grosspay[i] > 1000){

taxrate[i] = .3;

}

else if(grosspay[i] > 800){

taxrate[i] = .2;

}

else if(grosspay[i] > 500){

taxrate[i] = .1;

}

else{

taxrate[i] = 0;

}

// adjust tax rate based on status

if(status[i] == 's' || status[i] == 'S'){

taxrate[i] += .05;

}

else if(status[i] == 'h' || status[i] == 'H'){

taxrate[i] -= .05;

}

if(taxrate[i] < 0) {

taxrate[i] = 0;

}

// calculate tax and net pay

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

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

// print employee data

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

<< endl;

// increment the index number

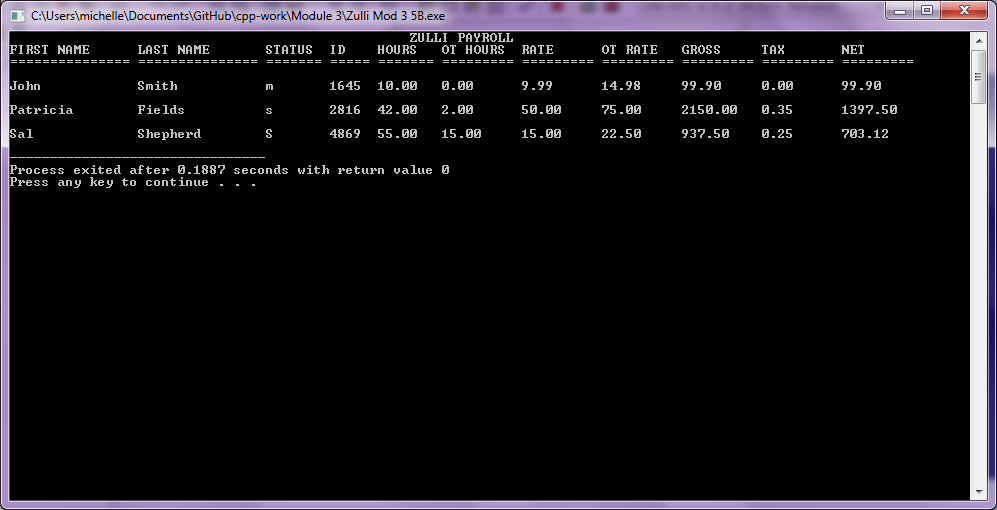
i++;

}//WHILE

return 0;

}//MAIN

## B: Arrays and Loops — Output



## C: Functions — Code

#include<iostream>

#include<fstream>

#include<iomanip>

using namespace std;

// function prototypes

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 findtaxrate(float[], float[], char[], int);

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

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

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

int main() {

const int SIZE = 3; // set array size

// declare arrays

string firstname[SIZE], lastname[SIZE];

int id[SIZE];

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

float taxamount[SIZE], taxrate[SIZE];

char status[SIZE];

// function calls

int n = readdata(firstname, lastname, status, id, hours, rate, SIZE);

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

findotrate(ot\_rate, rate, n);

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

findtaxrate(grosspay, taxrate, status, n);

findtaxamt(taxamount, grosspay, taxrate, n);

findnetpay(netpay, grosspay, taxamount, n);

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

return 0;

} // MAIN

// function definitions:

// pull data from the input file to the array

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

ifstream fin("employee5A.in");

n = 0;

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

fin.close();

return n;

} // readdata

// set base and ot hours

void 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 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 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 rate based on gross and status

void findtaxrate(float grosspay[], float taxrate[], char status[], int n){

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

if(grosspay[i] > 1000){

taxrate[i] = .3;

}

else if(grosspay[i] > 800){

taxrate[i] = .2;

}

else if(grosspay[i] > 500){

taxrate[i] = .1;

}

else{

taxrate[i] = 0;

}

// adjust tax rate based on status

if(status[i] == 's' || status[i] == 'S'){

taxrate[i] += .05;

}

else if(status[i] == 'h' || status[i] == 'H'){

taxrate[i] -= .05;

}

if(taxrate[i] < 0) {

taxrate[i] = 0;

}

}

} // findtaxrate

// calculate tax amount

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

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

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

}

} // findtaxamt

// calculate net pay

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

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

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

}

} // findnetpay

// print data table

void 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[], 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[i] << setw(10) << netpay[i]

<< endl;

}

} // printdata

// END

## C: Functions — Output

