Michelle Zulli

Dr. Alireza Ebrahimi

Introduction to C++ and OOP

June 13, 2015

# Module 2: Payroll

## 3A: Documentation

The payroll system was expanded to allow the user to set the number of employees he or she would like to enter, and then to loop the program the number of times the user set. The was accomplished by adding a variable called counter and setting it to the value that was entered by the user. The program then asks for the employee data, prints the entered information to the screen, and decrements the counter variable by 1. The loop terminates when the counter reaches 0.

## 3A: Code

#include<iostream>

using namespace std;

main() {

int id, counter;

float hours, rate, grosspay, netpay, taxamount;

float const TAXRATE = .10;

cout << "HOW MANY EMPLOYEES DO YOU WANT TO ENTER? ";

cin >> counter;

cout << endl;

while(counter > 0){

cout << "ENTER EMPLOYEE ID (LAST 4 DIGITS OF SS#): ";

cin >> id;

cout << "ENTER HOURS WORKED: ";

cin >> hours;

cout << "ENTER HOURLY RATE: ";

cin >> rate;

grosspay = hours \* rate;

taxamount = grosspay \* TAXRATE;

netpay = grosspay - taxamount;

cout << endl;

cout << "EMPLOYEE ID: " << id << endl;

cout << "HOURS WORKED: " << hours << endl;

cout << "HOURLY RATE: " << rate << endl;

cout << "GROSS PAY: $" << grosspay << endl;

cout << "TAX RATE: " << TAXRATE << endl;

cout << "TAX AMOUNT: $" << taxamount << endl;

cout << "NET PAY: $" << netpay << endl;

cout << endl;

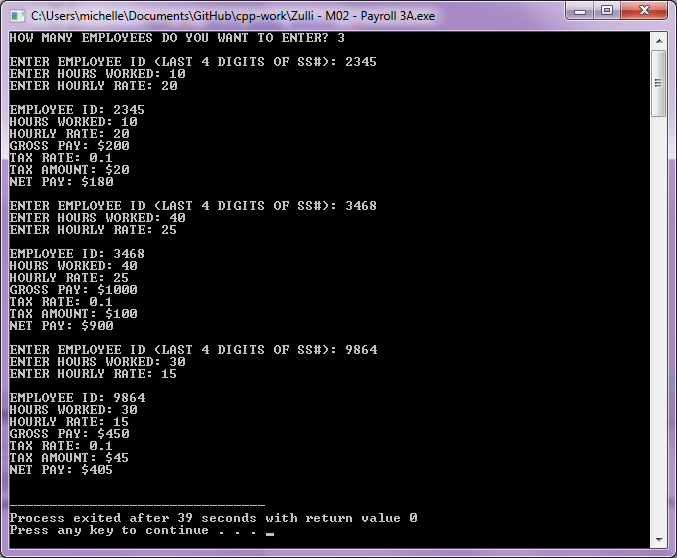
counter--;

}//WHILE

return 0;

}//MAIN

## 3A: Output



## 3B: Documentation

The payroll system was modified to use an input file and execute the loop for as many employees as are in the file. To do this, the an include statement was added for the fstream header file. Then, the counter variable was removed from the program and an ifstream object was created from the input file (“employee.in”). Finally, the loop condition was changed to loop through the input file assigning the values there to the program variables and terminate at end of file.

## 3B: Code

#include<iostream>

#include<fstream>

using namespace std;

main() {

int id;

float hours, rate, grosspay, netpay, taxamount;

float const TAXRATE = .10;

ifstream fin("employee.in");

while(fin >> id >> hours >> rate){

grosspay = hours \* rate;

taxamount = grosspay \* TAXRATE;

netpay = grosspay - taxamount;

cout << "EMPLOYEE ID: " << id << endl;

cout << "HOURS WORKED: " << hours << endl;

cout << "HOURLY RATE: " << rate << endl;

cout << "GROSS PAY: $" << grosspay << endl;

cout << "TAX RATE: " << TAXRATE << endl;

cout << "TAX AMOUNT: $" << taxamount << endl;

cout << "NET PAY: $" << netpay << endl;

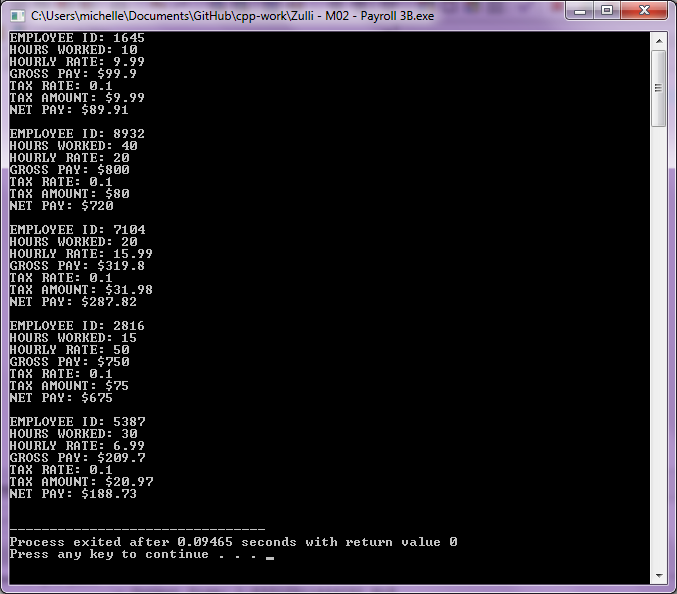
cout << endl;

}//WHILE

return 0;

}//MAIN

## 3B: Output



## 4A: Documentation

The payroll system was modified to take the employee’s tax status into account. Possible statuses are “S” for Single, “M” for Married, and “H” for head of household. The program gives a user error if the status is invalid. It uses a series of if statements to calculate the final tax rate. The program checks to make sure that no employee’s tax rate is less than zero before it calculates and outputs the results.

## 4A: Code

#include<iostream>

#include<iomanip>

using namespace std;

main() {

int id, counter;

float hours, rate, grosspay, netpay, taxamount, taxrate;

char status;

cout << "HOW MANY EMPLOYEES DO YOU WANT TO ENTER? ";

cin >> counter;

cout << endl;

while(counter > 0){

cout << "ENTER EMPLOYEE ID (LAST 4 DIGITS OF SS#): ";

cin >> id;

cout << "ENTER HOURS WORKED: ";

cin >> hours;

cout << "ENTER HOURLY RATE: ";

cin >> rate;

cout << "ENTER STATUS -- S for Single, M for Married, H for Head of Household: ";

cin >> status;

if((status != 's') && (status != 'S') && (status != 'm') && (status != 'M') && (status != 'h') && (status != 'H')){

cout << "INVALID - ENTER STATUS: ";

cin >> status;

}

grosspay = hours \* rate;

if(grosspay < 0){

cout << "ERROR!" << endl;

}

else{

if(grosspay > 1000){

taxrate = .3;

}

else if(grosspay > 800){

taxrate = .2;

}

else if(grosspay > 500){

taxrate = .1;

}

else{

taxrate = 0;

}

}

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

taxrate += .05;

}

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

taxrate -= .05;

}

if(taxrate < 0) {

taxrate = 0;

}

taxamount = grosspay \* taxrate;

netpay = grosspay - taxamount;

cout << setprecision(2)

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

<< endl

<< "EMPLOYEE ID: " << setw(12) << id << endl

<< "HOURS WORKED: " << setw(12) << hours << endl

<< "HOURLY RATE: " << setw(12) << rate << endl

<< "GROSS PAY: $" << setw(12) << grosspay << endl

<< "STATUS: " << setw(12) << status << endl

<< "TAX RATE: " << setw(12) << taxrate << endl

<< "TAX AMOUNT: $" << setw(12) << taxamount << endl

<< "NET PAY: $" << setw(12) << netpay << endl

<< endl;

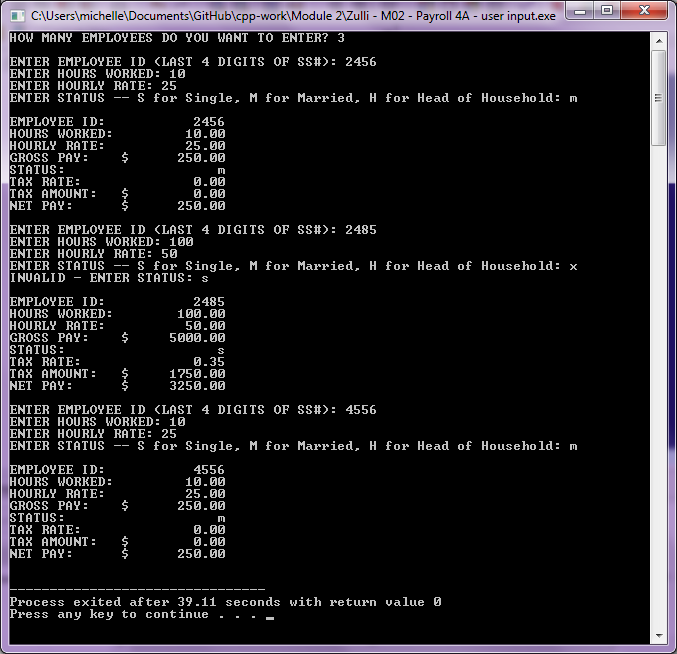
counter--;

}//WHILE

return 0;

}//MAIN

## 4A: Output



## 4B: Documentation

The payroll system was further modified to calculate overtime hours. The entered hours were split into two variables, base\_hours and ot\_hours. The entered pay rate was used to generate an overtime rate of 1.5 times the base rate. The program calculates gross pay by calculating base\_hours times base\_rate and adding ot\_hours multiplied by ot\_rate.

## 4B: Code

#include<iostream>

#include<iomanip>

using namespace std;

main() {

int id, counter;

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

char status;

cout << "HOW MANY EMPLOYEES DO YOU WANT TO ENTER? ";

cin >> counter;

cout << endl;

while(counter > 0){

cout << "ENTER EMPLOYEE ID (LAST 4 DIGITS OF SS#): ";

cin >> id;

cout << "ENTER HOURS WORKED: ";

cin >> hours;

if(hours <= 40) {

base\_hours = hours;

ot\_hours = 0;

}

else {

base\_hours = 40;

ot\_hours = hours - 40;

}

cout << "ENTER HOURLY RATE: ";

cin >> rate;

ot\_rate = rate \*1.5;

cout << "ENTER STATUS -- S for Single, M for Married, H for Head of Household: ";

cin >> status;

if((status != 's') && (status != 'S') && (status != 'm') && (status != 'M') && (status != 'h') && (status != 'H')){

cout << "INVALID - ENTER STATUS: ";

cin >> status;

}

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

if(grosspay < 0){

cout << "ERROR!" << endl;

}

else{

if(grosspay > 1000){

taxrate = .3;

}

else if(grosspay > 800){

taxrate = .2;

}

else if(grosspay > 500){

taxrate = .1;

}

else{

taxrate = 0;

}

}

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

taxrate += .05;

}

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

taxrate -= .05;

}

if(taxrate < 0) {

taxrate = 0;

}

taxamount = grosspay \* taxrate;

netpay = grosspay - taxamount;

cout << setprecision(2)

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

<< endl

<< "EMPLOYEE ID: " << setw(12) << id << endl

<< "TOTAL HOURS WORKED: " << setw(12) << hours << endl

<< "OT HOURS: " << setw(12) << ot\_hours << endl

<< "BASE RATE: $" << setw(12) << rate << endl

<< "OT RATE: $" << setw(12) << ot\_rate << endl

<< "GROSS PAY: $" << setw(12) << grosspay << endl

<< "STATUS: " << setw(12) << status << endl

<< "TAX RATE: " << setw(12) << taxrate << endl

<< "TAX AMOUNT: $" << setw(12) << taxamount << endl

<< "NET PAY: $" << setw(12) << netpay << endl

<< endl;

counter--;

}//WHILE

return 0;

}//MAIN

## 4B: Output

