CSCI 240

Qiguang Yang

Jordan Ringenberg

10/14/2015

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Name: Qiguang Yang

Date: 10/10/2015

Course: CSCI240

Properties: This program allow mutiple users to inpout saleries in different years.

Then the taxes of each user's each year will be print.

Inputs: Name, SSN, income

Outputs: Name, SSN, income, income tax and tax rate.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <iostream>

#include <string>

#include <fstream>

#include <iomanip>

using namespace std;

int main(){

const int MaxSize = 20;

string Name;

int SSN;

int Income[MaxSize];

int year[MaxSize];

int numIncome = 0;

int numYear = 0;

int numTaxBrackets = 0;

bool flag1 = 0;

bool flag2 = 0;

double code = 0;

int brackets = 0;

double Tax[MaxSize];

double taxRate[MaxSize];

double taxBracketPct[MaxSize][MaxSize];

double taxBracketFloor[MaxSize][MaxSize];

ifstream inTax("tax\_brackets.dat");

ifstream inCustomer("customer3\_tax.dat");

ofstream outFile("tax\_receipt.txt");

while(code != -2){

inTax >> code;

if(code == -2){

inTax.ignore(256,'\n');

flag1 = 1;

break;

}

year[numYear] = code;

while(code != -1){

inTax >> code;

if(code == -1){

inTax.ignore(256,'\n');

flag2 = 1;

break;

}

taxBracketPct[numYear][numTaxBrackets] = code;

inTax >> taxBracketFloor[numYear][numTaxBrackets];

numTaxBrackets ++;

brackets = numTaxBrackets;

}

flag2 = 0;

numTaxBrackets = 0;

numYear ++;

}

getline(inCustomer,Name);

inCustomer >> SSN;

while(inCustomer >> Income[numIncome]){

if(numIncome >= numYear){

break;

}

numIncome ++;

}

double taxLevel[MaxSize][MaxSize];

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

for(int j = 0; j < brackets; j++){

if(j == 0){

taxLevel[i][0] = taxBracketFloor[i][1] \* taxBracketPct[i][0];

}

else{

taxLevel[i][j] = taxLevel[i][j-1] + (taxBracketFloor[i][j+1] - taxBracketFloor[i][j]) \* taxBracketPct[i][j];

}

}

}

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

if(year[i] <= 2001){

brackets -= 1;

}

for(int j =0; j < brackets ; j++){

if(Income[i] < taxBracketFloor[i][1]){

Tax[i] = Income[i] \* taxBracketPct[i][0];

}

if(Income[i] > taxBracketFloor[i][brackets]){

Tax[i] = (Income[i] - taxBracketFloor[i][brackets]) \* taxBracketPct[i][brackets] + taxLevel[i][j];

}

else if(Income[i] > taxBracketFloor[i][j+1] && Income[i] < taxBracketFloor[i][j+2]){

Tax[i] = (Income[i] - taxBracketFloor[i][j+1]) \* taxBracketPct[i][j+1] + taxLevel[i][j];

}

}

taxRate[i] = Tax[i] / Income[i] \* 100;

}

outFile << setw(50) << setfill('\*') << "\*" << endl;

outFile << "|Taxpayer Name:" << setw(34) << setfill('.');

outFile << Name << "|" << endl;

outFile << "|Taxpayer SSN:" << setw(26) << setfill('.') << ".";

outFile << setw(9) << setfill('0') << SSN << "|" << endl;

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

outFile << setw(50) << setfill('-') << "-" << endl;

outFile << "|Year:" << setw(43) << setfill('.');

outFile << year[i] << "|" << endl;

outFile << "|Income ($):" << setw(37) << setfill('.');

outFile << fixed << setprecision(0) << Income[i] << "|" << endl;

outFile << "|Taxes Owed ($):" << setw(33) << setfill('.');

outFile << fixed << setprecision(0) << Tax[i] << "|" << endl;

outFile << "|Effective Rate:" << setw(32) << setfill('.');

outFile << fixed << setprecision(1) << taxRate[i] << "%|" << endl;

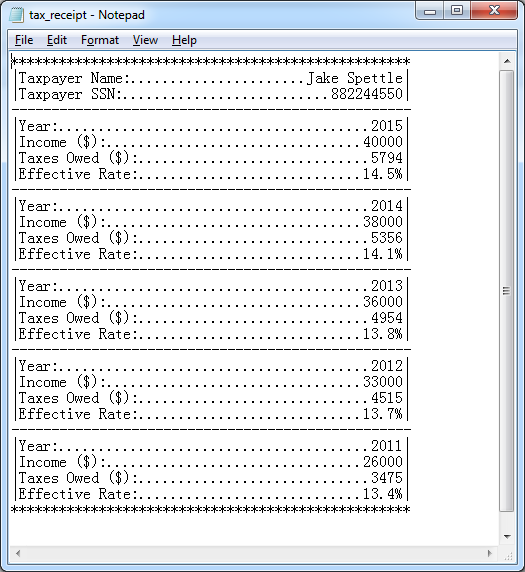
}

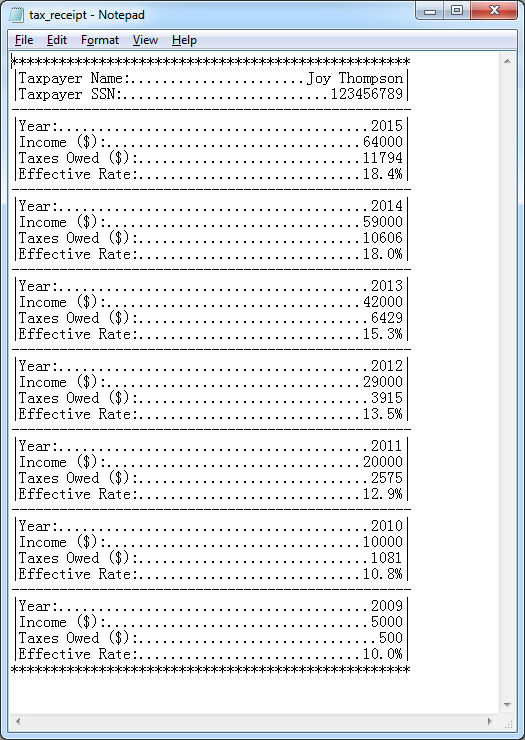
outFile << setw(50) << setfill('\*') << "\*" << endl;

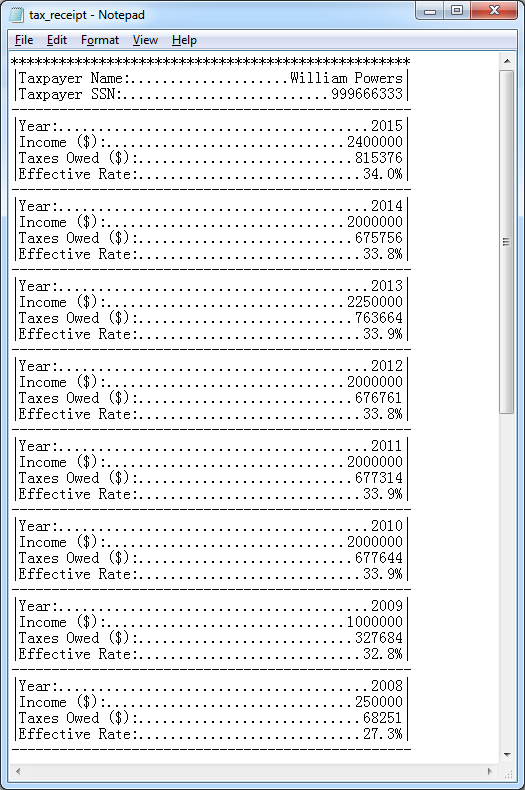
outFile << endl;

return 0;

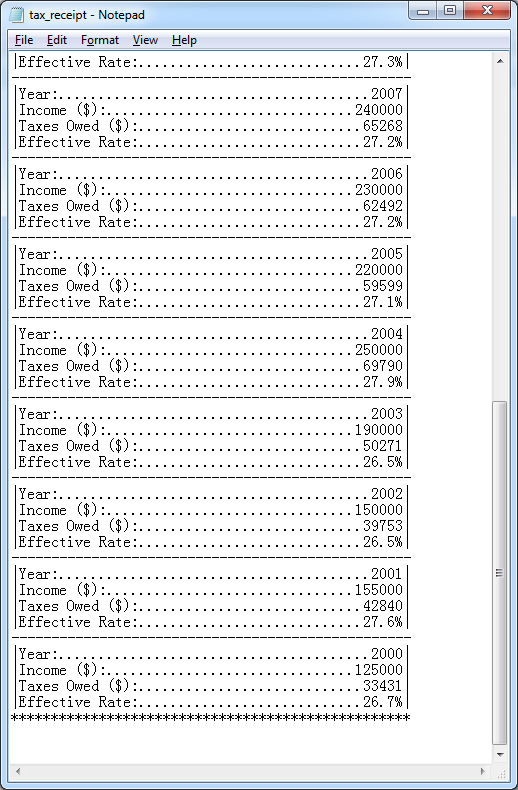
}







Continued

.