Jessica Sullivan

ID: 1282151

Class: COMSC-210-5067

Professor: Pentcheva

Source Code:

CS210\_Assignment1A.cpp:

*/\**

*Programmer: Jessica Sullivan*

*Programmer's ID: 1282151*

*Class: COMSC-210-5067*

*Goal: Write a program that will record information about employees*

*and will compute their paychecks*

*\*/*

#include <string>

#include <sstream>

#include <iostream>

#include <iomanip>

**using** **namespace** std;

#include "Employee.h"

*// Listing of constants for easy changes to 7 day weeks or other*

*// adjustments*

**static** **const** **int** NUM\_EMPLOYEES = 4;

**static** **const** **int** NUM\_DAYS = 5;

**static** **const** **char**\* DAYS[] = {"MONDAY", "TUESDAY", "WEDNESDAY",

"THURSDAY", "FRIDAY"};

**static** **const** **double** NUM\_HOURS\_FOR\_OVERTIME = 40;

**static** **const** **double** PERCENT\_WAGE\_OT = 1.5;

**bool** validateInput(**const** string &input, **double** &value);

**void** initialize(Employee \*employees); *// initialize members of employee*

**void** compute(Employee &employee); *// calculate total wages*

**void** result(**const** Employee &employee); *// prints employee info*

**int** main(**int** argc, **const** **char** \* argv[]) {

Employee employees[NUM\_EMPLOYEES];

initialize(employees);

**for** (**int** i = 0; i < NUM\_EMPLOYEES; i++) {

compute(employees[i]);

result(employees[i]);

}

**return** 0;

}

*// ensures that number input is valid - no whitespace, letters, or negatives*

**bool** validateInput(**const** string &input, **double** &value) {

stringstream ss;

ss.str(input);

ss >> value;

**return** value >= 0 && !ss.fail();

}

*// gets info from users and calls validateInput on numbers inputed*

**void** initialize(Employee \*employees) {

string input, lName = "";

**double** hours = -1, wages = -1;

stringstream ss;

**bool** valid = **false**;

*// go through each of the employees to get info*

**for** (**int** i = 0; i < NUM\_EMPLOYEES; i++) {

cout << endl << "Enter Last Name of Employee " << i + 1 << ": ";

getline(cin, lName);

employees[i].mLName = lName; *// no error catching required*

*// go through each day of the work week*

**for** (**int** j = 0; j < NUM\_DAYS; j++) {

**do** {

cout << endl << "Enter hours worked on: "

+ **static\_cast**<string>(DAYS[j]) + " : ";

getline(cin, input);

*//valid makes sure input is valid for hours*

valid = validateInput(input, hours) && hours <= 24;

**if** (!valid) {

cerr << endl << "Error, " << input << " is an invalid input."

<< endl;

hours = -1;

}

input = "";

} **while** (!valid);

employees[i].mHours[j] = hours;

valid = **false**;

}

**do** {

cout << endl << "Enter hourly wage : $";

getline(cin, input);

*//valid makes sure wages are valid*

valid = validateInput(input, wages);

**if** (!valid) {

cerr << endl << "Error, " << input << " is an invalid input."

<< endl;

wages = -1;

input = "";

}

} **while** (!valid);

employees[i].mHourlyWage = wages;

}

}

*// computes wages of employees based on total hours worked and wage rate*

**void** compute(Employee &employee) {

**double** totalHours = 0;

**for** (**int** i = 0; i < NUM\_DAYS; i++) {

totalHours += employee.mHours[i];

}

employee.mWages = totalHours > NUM\_HOURS\_FOR\_OVERTIME

? NUM\_HOURS\_FOR\_OVERTIME \* employee.mHourlyWage

+ (totalHours - NUM\_HOURS\_FOR\_OVERTIME)

\* employee.mHourlyWage \* PERCENT\_WAGE\_OT

: totalHours \* employee.mHourlyWage;

}

*//prints out relevant information about employees.*

**void** result(**const** Employee &employee) {

cout << endl << "Employee Name: " << employee.mLName << endl;

cout << endl << "Hours Worked: " << endl;

**for** (**int** i = 0; i < NUM\_DAYS; i++) {

cout << setw(10) << left << DAYS[i] << ": " << setw(5) << right

<< employee.mHours[i] << " hours" << endl;

}

cout << endl << "Total Wages: $" << fixed << setprecision(2)

<< employee.mWages << endl;

}

Employee.h:

*//*

*// Employee.h*

*// CS210\_Assignment1A*

*//*

*// Created by Jessie Sully on 1/28/20.*

*// Copyright © 2020 Jessie Sully. All rights reserved.*

*//*

#ifndef Employee\_h

#define Employee\_h

**struct** Employee {

**static** **const** **int** NUM\_WORKDAYS = 5;

string mLName;

**double** mHours[NUM\_WORKDAYS];

**double** mHourlyWage;

**double** mWages;

};

#endif */\* Employee\_h \*/*

Output:







