

**Iteration & calculations:****Payroll Program**

This project will build on some of the functionality that you were required to implement for Project 2. The major difference is that now your program must be able to input data for multiple employees and output the payroll information for all of the employees to a single file.

The input data will be in a different format. For each employee, you will be given the employee's name, the hours worked, a code indicating the employee's Base Pay Rate, and a code indicating the Health Insurance plan.

The Pay Code shall be interpreted as follows:

Pay Code	Base Pay Rate
M	17.50
C	9.00
L	8.50

The Gross Pay will take into account overtime hours and is to be calculated in the following manner:

- For each of the first 40 hours, an employee is paid at a rate of the Base Pay Rate indicated by the Pay Code.
- For each of the next 10 hours, an employee is paid at a rate of 1.5 times the Base Pay Rate.
- For each additional hour beyond the first 50 hours, an employee is paid at a rate of 2.0 times the Base Pay Rate.

For example, suppose an employee with a Base Pay Rate of \$9.00 worked 52 hours. Her Gross Pay would be calculated as follows:

$$\begin{aligned} \text{Gross Pay} &= 40 * 9.00 \\ &+ 10 * 1.5 * 9.00 \\ &+ 2 * 2.0 * 9.00; \end{aligned}$$

Note: The use of 9.00 in the equation above is for illustrative purposes only. Your code should use 900 to represent \$9.00.

The FIT withholding will be calculated using a "bracketed" approach (similar to how gross pay is calculated). The tax rate to be applied depends upon the employee's gross pay amount, as shown in the table below:

Bracket	FIT Rate
0.00 to 350.00	10%
350.01 to 500.00	15%
500.01 & up	27%

As in Project 2, the Social Security Old Age tax (FICA) and the Medicare tax (FMED) must still be deducted. The rate for the FICA tax is 6.20%, and the rate for the FMED tax is 1.45%. In addition, the State Income Tax (SIT) rate will be fixed at 5% of the gross pay amount (regardless of what the gross pay is).

The Health Insurance Premium will depend upon which of three plans an employee has selected. This will be indicated by an Insurance Plan Code, corresponding to the following premiums:

Insurance Plan Code	Premium
A	32.75
B	20.00
C	0.00

(Plan C indicates that the employee has opted not to purchase the employer's health insurance.)

**Sample input data:**

Here is a sample input file, named `PayData.txt`, for the program

Snow White and the 5 Dwarfs			
Name	Hours	Pay	Ins
Dwarf, Sleepy	54.3	L	B
Dwarf, Doc	37.4	C	A
Dwarf, Happy	49.4	L	A
White, Snow	35.6	M	C
Dwarf, Grumpy	40.1	C	B
Dwarf, Dopey	47.9	L	A

The first line contains the company name, which should be saved for the output. The next three lines should be ignored by your program. Each of the remaining lines contains four pieces of payroll data for a single employee:

- employee name                    a string of no more than 15 characters, containing no tabs
- hours worked                    a nonnegative decimal number
- pay rate code                    one of the characters 'M', 'L', or 'C'
- insurance plan code            one of the characters 'A', 'B', 'C'

There will be a single tab separating these items, and all are guaranteed to be logically correct. A newline character will immediately follow the insurance plan code.

There is no specified limit on the number of employee data lines, so your program must be designed to detect when it's out of input and stop automatically.

**Calculations:**

As before, you must calculate the gross pay, the withholding amount for all the taxes, the insurance premium, and finally the net pay.

In addition, remember that in order to produce the same results as the Curator, all monetary amounts must use integers.

**Sample output:**

Here is a sample output file, named `PayStub.txt`, which corresponds to the input data given above:

Programmer: Rich Wheaton							
CS 1044 Project 3 Summer II 2003							
Company: Snow White and the 5 Dwarfs							
Employee Name	Gross	FIT	FMED	FICA	SIT	Ins	Net
Dwarf, Sleepy	540.59	68.45	7.83	33.51	27.02	20.00	383.78
Dwarf, Doc	336.60	33.66	4.88	20.86	16.83	32.75	227.62
Dwarf, Happy	459.84	51.47	6.66	28.51	22.99	32.75	317.46
White, Snow	623.00	90.71	9.03	38.62	31.15	0.00	453.49
Dwarf, Grumpy	361.35	36.70	5.23	22.40	18.06	20.00	258.96
Dwarf, Dopey	440.72	48.60	6.39	27.32	22.03	32.75	303.63

The first two lines just identify the programmer and project, and the third line is a blank. The next line contains the company name from the input file. The fifth line is blank, and the sixth line contains the column headings for the output table. Next comes a separator line. Next there is a line for each employee, with all values output to precision 2, in the same order as the employees specified in the input file. The information for each employee is shown in the sample output file on the previous page. Finally, there is a separator line after the employee data.

If you have read the description of how the Curator scores your program in the *Student Guide*, you know that is important that you use the same spelling and capitalization for all the labels shown above. The horizontal spacing does not effect scoring unless you combine things that should be separate or separate things that should be combined. You are free to experiment with the horizontal layout but you should try to align the columns neatly.

### **Evaluation:**

Everything that you have been told about testing in class applies here. Do not waste submissions to the Curator in testing your program! There is no point in submitting your program until you have verified that it produces correct results on the sample data files that are provided. If you waste all of your submissions because you have not tested your program adequately then you will receive a low score on this assignment. You will not be given extra submissions.

Your submitted program will be assigned a score, out of 100, based upon the runtime testing performed by the Curator System. We will also be evaluating your submission of this program for documentation style and a few good coding practices. This will result in a deduction (ideally zero) that will be applied to your score from the Curator to yield your final score for this project.

Read the *Programming Standards* page on the CS 1044 website for general guidelines. You should comment your code in the same manner as the code given for the first two programming assignments. In particular:

- You should have a header comment identifying yourself, and describing what the program does.
- Every constant and variable you declare should have a comment explaining its logical significance in the program.
- Every major block of code should have a comment describing its purpose.
- Adopt a consistent indentation style and stick to it.

Your implementation must also meet the following requirements:

- Choose descriptive identifiers when you declare a variable or constant. Avoid choosing identifiers that are entirely lower-case.
- Use named constants instead of literal constants when the constant has logical significance.
- Use C++ streams for input and output, not C-style constructs.
- Use C++ string variables to hold character data, not C-style character pointers or arrays.
- Note: you are explicitly forbidden to write any user-defined functions for this program. This may make the program somewhat repetitive, and physically longer than the alternative. To some extent, that's the reason for this restriction.

Understand that the list of requirements here is not a complete repetition of the *Programming Standards* page on the course website. It is possible that requirements listed there will be applied, even if they are not listed here.

### **Submitting your program:**

You will submit this assignment to the Curator System (read the *Student Guide*), and it will be graded automatically. Instructions for submitting, and a description of how the grading is done, are contained in the *Student Guide*.

You will be allowed up to five submissions for this assignment. Use them wisely. Test your program thoroughly before submitting it. Make sure that your program produces correct results for every sample input file posted on the course website. If you do not get a perfect score, analyze the problem carefully and test your fix with the input file returned as part of the Curator e-mail message, before submitting again. The highest score you achieve will be counted.

The *Student Guide* and submission link can be found at:

<http://www.cs.vt.edu/curator/>

### **Pledge:**

Each of your program submissions must be pledged to conform to the Honor Code requirements for this course. Specifically, you **must** include the following pledge statement in the header comment for your program:

```
// On my honor:  
//  
// - I have not discussed the C++ language code in my program with  
//   anyone other than my instructor or the teaching assistants  
//   assigned to this course.  
//  
// - I have not used C++ language code obtained from another student,  
//   or any other unauthorized source, either modified or unmodified.  
//  
// - If any C++ language code or documentation used in my program  
//   was obtained from another source, such as a text book or course  
//   notes, that has been clearly noted with a proper citation in  
//   the comments of my program.  
//  
// - I have not designed this program in such a way as to defeat or  
//   interfere with the normal operation of the Curator System.  
//  
// <Student Name>
```

**Failure to include this pledge in a submission is a violation of the Honor Code.**