**Common Project**

**Name: Student Full Name**

**Instructor: Professor Full Name**

**Class: CMSC 140**

**Course CRN:**

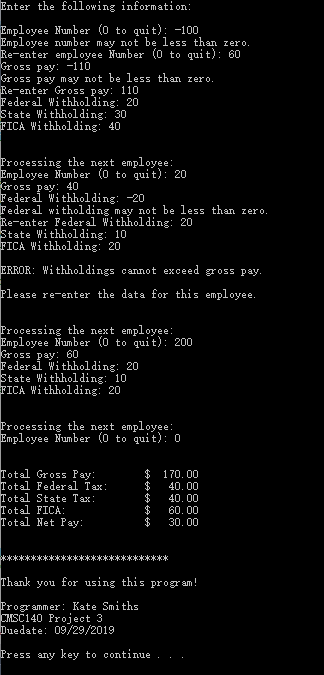
**Due Date: mm/dd/yyyy**

**Test Plan (**at least 3 test cases.As a programmer, you are responsible for your source code covers **all scenarios possible in the scope of concepts** described in Chapters of this and previous modules/weeks**)**.

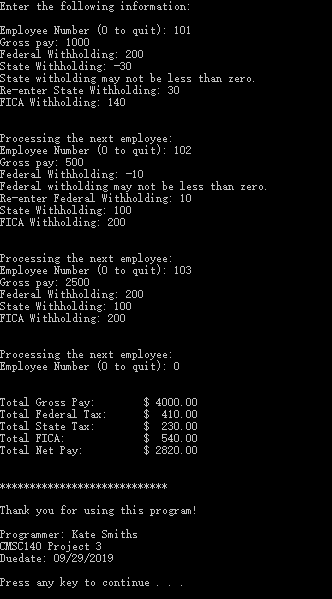
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Cases** | **Input** | **Expected Output** | **Actual Output** | **Did the Test Pass?** |
| **Case 1** | Employee Number (0 to quit): -100 Re-enter employee Number (0 to quit): 60 Gross pay: -110  Re-enter gross pay: 110  Federal Witholding: 20  State Witholding: 30  FICA Witholding: 40  Processing the next employee:  Employee Number (0 to quit): 200  Gross pay: 40  Federal Witholding: -20  Re-enter Federal Witholding: 20  State Witholding: 10  FICA Witholding: 20  Employee Number (0 to quit): 200  Gross pay: 60  Federal Witholding: 20  State Witholding: 10  FICA Witholding: 20  Employee Number (0 to quit): 0 | Employee number may not be less than zero.  Employee number may not be less than zero.  Gross pay may not be less than zero.  Federal witholding may not be less than zero.  ERROR: Withholdings cannot exceed gross pay.  Please re-enter the data for this employee.  Total Gross Pay: $ 170.00  Total Federal Tax: $ 40.00  Total State Tax: $ 40.00  Total FICA: $ 60.00  Total Net Pay: $ 30.00  Programmer: Kate Smiths  CMSC140 Project 3  Duedate: 09/29/2019 | Employee number may not be less than zero.  Employee number may not be less than zero.  Gross pay may not be less than zero.  Federal witholding may not be less than zero.  ERROR: Withholdings cannot exceed gross pay.  Please re-enter the data for this employee.  Total Gross Pay: $ 170.00  Total Federal Tax: $ 40.00  Total State Tax: $ 40.00  Total FICA: $ 60.00  Total Net Pay: $ 30.00  Programmer: Kate Smiths  CMSC140 Project 3  Duedate: 09/29/2019 | Y |
| **Case 2** | Employee Number (0 to quit): 101 Gross pay: 1000  Federal Witholding: 200  State Witholding: -30  Re-enter state withholding: 30  FICA Witholding: 140  Employee Number (0 to quit): 102  Gross pay: 500  Federal Witholding: -10  Re-enter Federal Witholding: 10  State Witholding: 100  FICA Witholding: 200  Employee Number (0 to quit): 103  Gross pay: 2500  Federal Witholding: 200  State Witholding: 100  FICA Witholding: 200  Employee Number (0 to quit): 0 | Not be less than zero.  Not be less than zero.  State witholding may not be less than zero.  Not be less than zero.  Federal witholding may not be less than zero.  ERROR: Withholdings cannot exceed gross pay.  Federal may not be less than zero.  Total Gross Pay: $ 4000.00  Total Federal Tax: $ 410.00  Total State Tax: $ 230.00  Total FICA: $ 540.00  Total Net Pay: $ 2820.00  Programmer: Kate Smiths  CMSC140 Project 3  Duedate: 09/29/2019 | Not be less than zero.  Not be less than zero.  State witholding may not be less than zero.  Not be less than zero.  Federal witholding may not be less than zero.  ERROR: Withholdings cannot exceed gross pay.  Federal may not be less than zero.  Total Gross Pay: $ 4000.00  Total Federal Tax: $ 410.00  Total State Tax: $ 230.00  Total FICA: $ 540.00  Total Net Pay: $ 2820.00  Programmer: Kate Smiths  CMSC140 Project 3  Duedate: 09/29/2019 | Y |
| **Case 3** | Employee Number (0 to quit): 111  Gross pay: 6000  Federal Witholding: 500  State Witholding: 400  FICA Witholding: 300  Employee Number (0 to quit): 0 | Total Gross Pay: $ 6000.00  Total Federal Tax: $ 500.00  Total State Tax: $ 400.00  Total FICA: $ 300.00  Total Net Pay: $ 4800.00  Programmer: Kate Smiths  CMSC140 Project 3  Duedate: 09/29/2019 | Total Gross Pay: $ 6000.00  Total Federal Tax: $ 500.00  Total State Tax: $ 400.00  Total FICA: $ 300.00  Total Net Pay: $ 4800.00  Programmer: Kate Smiths  CMSC140 Project 3  Duedate: 09/29/2019 | Y |

**Screenshot of the program run**

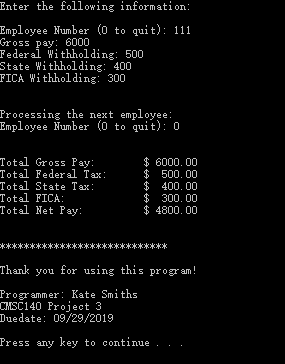
**Test 1:**



**Test 2:**



**Test 3:**



**Pseudocode / Algorithm for the program**

Start

Define variables, and initial some variables.

Prompt the user for the Employer Number (empNum)

Read empNum

While (empNum < 0)

Display error message and prompt user to re-enter empNum

Read empNum

While (empNum != 0)

Prompt the user for the gross pay (gross)

Read gross

While ( gross < 0.0 )

Display error message and prompt user to re-enter gross

Read gross

Prompt the user for the federal withholding (fedTax)

Read fedTax

While ( fedTax < 0.0 )

Display error message and prompt user to re-enter fedTax

Read fedTax

Prompt the user for the state withholding (stateTax)

Read stateax

While ( stateTax < 0.0 )

Display error message and prompt user to re-enter stateTax

Read stateTax

Prompt the user for the FICA withholding (fica)

Read fica

While ( fica < 0.0 )

Display error message and prompt user to re-enter fica

Read fica

if ( (fedTax + stateTax + fica) > gross )

Display error message and prompt user to re-enter this employee’s data

else

totalGross += gross;

totalFed += fedTax;

totalState += stateTax;

totalFica += fica;

Prompt user for the empNum of the next employer

Read empNum

While ( empNum < 0)

Display error message and prompt user to re-enter empNum

Read empNum

Calculation: totalNet = totalGross - totalFed - totalState - totalFica;

Display the calculation results using setw manipulator

Display the programmer’s name, project’s name and due date

End

**Flowchart**

**(**Follow the textbook App D Intro to Flowcharting instruction for drawing the flowchart**)**

**Lessons learned (Example):**

I learned a lot from this project. Firstly, completing the project helps me to review what I have learned, including variable initialization, type casting expression and getline function, from this course until now. This makes me form a solid basis in programming. Moreover, I learned that programmer should always take programming seriously and carefully, because even a tiny punctuation would cause a completely different result. For example, when using the type casting **“static\_cast< *DataType* >( *Value* )”**, only “num1” but not the whole mathematical expression “num1/num2” is included in the parentheses. Otherwise, the integer division will happen prior to the data type conversion.

During my work on this project, I met with some problems. The first problem is the syntax errors. When I compiled the source code, it showed that it failed. However, fortunately, the software would automatically give me some hints that I missed some semicolons in the error lists. Then I could locate the exact line and fix them. The other problem is that although the program ran successfully at first, however, when I typed a full name with spaces in between, such as Mary Lee, the program just showed all the following dialogues without giving me chances to input the remaining variable values. Then, I turned to the textbook for help and found out that if I want to input a line, I have to apply getline function. Therefore, I modified the statement and it finally worked!

For this project, I worked it out on the due day because. It is a bit rush for me. I would start the second project as soon as possible, so that I could have more time to review the teaching materials and better performance in it.

**Check List:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nr** |  | **Y/N** | **Comments** |
| **1** | **Compressed files:** | **Y** |  |
|  | KSmiths\_Pr3\_Moss.zip | **Y** |  |
|  | KSmiths\_Pr3.zip | **Y** |  |
| **2** | **Source .cpp files** | **Y** |  |
| **3** | **Program compiles** | **Y** |  |
| **4** | **Documentation files:** | **Y** |  |
|  | **Comprehensive Test Plan** | **Y** |  |
|  | **Screenshots based on Test Plan** | **Y** |  |
|  | **Algorithms(Pseudocode)** | **Y** |  |
|  | **Flowchart** | **Y** |  |
|  | **Lessons Learned** | **Y** |  |
| **5** | **Program runs with desired outputs related to a Test Plan** | **Y** |  |