**Programming Assignment 2**

File IO

Control structures

Looping

Operator precedence

Error checking (bad data, filestream fail state, etc.)

Use of proper data types; type coercion and/or explicit type conversion

1. Write a program that can be used to calculate the federal tax obligation (amount of taxes owed) using the following information:

* For a single person, the standard exemption is $3,000
* For married people, the standard exemption is $6,000
* Personal exemption is $700 per person
* A person may put UP TO 5% of his/her gross (total income before taxes) in a pension plan. This can be 0, but not over 5%

Tax rates/amounts are:

* 15% - Taxable Income between $0 and $20,000
* $2,250 plus 25% of taxable income **OVER** $20,000 – Taxable Income between $20,001 and $50,000
* $8,460 plus 35% of taxable income **OVER** $50,000

1. Calculate the taxable income as follows:

Subtract the sum of (1) the standard exemption, (2) the amount contributed to a pension fund, and (3) the personal exemption **from** the Gross income. For example, if a married couple has two (2) children, personal exemption is 4 x $700 = $2,800. If a single person has 1 child, personal exemption is 2 \* $700 = $1,400.00.

Do NOT include the pension as an exemption (in the exemption total), include it as a deduction.

1. Tax obligation (taxes owed) will be calculated against the taxable income using the Tax Rates indicated above.
2. Request the name of the input file from the user. Do not “hard-code” the file name in the program.
3. Check the state of all file opens. If any file does not open properly, display an appropriate message and exit the program.
4. Read the file using a looping construct that checks for the end of file (not the number of rows in the data file).
5. Evaluate the marital status using a SWITCH statement.
6. Use the provided data file with the following information (inputFile.txt).

Each ROW of data contains:

Surname of family / single individual (last name only)

Marital status – represented by a character (M, S)

Number of children under the age of 14 – may be 0

Gross salary (before taxes)

Percentage of gross salary contributed to pension fund. This may be 0. This will be a percentage, not an amount, i.e., .05 will mean 5%.

1. For each row of input in the data file, output the following to the **CONSOLE** (screen) in a neat, readable format. Output each input line on one output line. Use manipulators to output values in readable columns. Use dollar signs before monetary values. All numerical values must have 2 decimal points. Use headings to indicate values displayed (you DO NOT need to use the following as headings, you may use your own):

Surname of the family (last name only)

Gross Income

Total exemption amount (does not include pension deduction)

Pension deduction amount in dollars (not the percentage)

Taxable Income

Tax obligation (taxes owed)

1. For each row of input in the data file, output the following to an **OUTPUT** **FILE,** name of your choice (in addition to the console) in a neat, readable format. Output each input line on one output line. Use manipulators to output values in readable columns. Use dollar signs before monetary values. All numerical values must have 2 decimal points. Use headings to indicate values displayed (you DO NOT need to use the following as headings; you may user your own).

You MAY hardcode the name of the output file, however, check the state of the open of the output file. If the open fails, output an appropriate message and exit the program:

Surname of the family (last name only)

Gross Income

Total exemption amount (does not include pension deduction)

Pension deduction amount in dollars (not the percentage)

Taxable Income

Tax obligation (taxes owed)

For 9 and 10, output must be in the following form, i.e., one heading for all lines for output:

Name Gross Income Total Exemptions Pension Deduction Taxable Income Tax Obligation

You MAY use abbreviations for headers to fit across the screen.

1. There will be ERRORS in the data file (see below for possible errors). If an error in the data is encountered, write the error and the line containing the error to a THIRD OUTPUT FILE, name of your choice (not the console). **DO NOT PROCESS THE DATA**; continue with the next input line. Only log the first error found in each line/row of data.

You MAY hardcode the name of the output error file, however, check the state of the open of the output file. If the open fails, output an appropriate message and exit the program:

* Amount contributed to a pension fund MAY NOT be OVER .05. If the value in the input file is OVER .05, output the message “Invalid Pension Percentage” and the line of data containing the error on a separate line. No headings.
* Marital Status must be ‘S’ or ‘M’. If any other value, output the message “Invalid Marital Status” and the line of data containing the error on a separate line. No headings.

1. CLOSE ALL files before exiting the program.

**Test and evaluate calculations for accuracy. Points will be taken for inaccurate calculations, improper formatting, directions not followed, no looping construct, no switch, etc.**

Turn in file code \*.cpp; README file; input file (even though provided) and ALL output files. You DO NOT need to submit the executable (\*.exe). You MAY zip all the files and submit if you choose. You MUST include all the files indicated or points will be deducted.

README must contain instructions for location of input/output files

Each solution is to be uniquely your own; no student or outside collaboration allowed.

**Points WILL BE taken if MINIMUM requirements and submissions not included.**

**I’ll award 1 point extra credit if the submission files are zipped and the zip file is submitted.**

See Blackboard and Syllabus for due date