CMPT220 – Prog2

Program Due: Monday, Feb. 10, before 9:00 a.m. (Submitted & Printed)

On my desk (in the classroom) beginning of class

Name the project **Prog2YourLastName**

Name the class **TaxesYourLastName**

The purpose of this program is to master conditional statements and loops. This program will compute taxes based on the following rules.

1) Taxable income is equal to gross income minus the standard deduction minus $1,200 for each exemption. The standard deduction is $3,250. The maximum number of allowable exemptions is 10. Remember that **taxes are not owed when taxable income is less than zero**. (If taxable income is less than zero, then no taxes are owed.)

2) The tax rate is computed from the following table based on filing status and taxable income:

Filing status	**Taxable income**	**Tax rate (for entire amount)**
Individual	less than $17,000	11%
	$17,000 - $58,000	20%
	more than $58,000	31%
Married filing **J**ointly	less than $20,000	14%
	$20,000 - $110,000	22%
	more than $110,000	39%
Head of household	less than $34,000	15%
	$34,000 - $75,000	23%
	more than $75,000	38%

3) The total tax due is the taxable income times the tax rate.

For a taxpayer, your program asks for the following (**in this order!**):

Taxpayer ID -- any integer, except 0,

* Filing Status – single character, (*do* allow lowercase, too)

I for Individual,

J for Married filing Jointly,

H for Head of household

Gross Income -- any double (negative allowed, representing a loss)

* Number of Exemptions -- an integer between 0 and 10, inclusive.

* means that you must validate the input. If invalid input is given, ask for it again (until it is valid).

The program computes the taxes owed and prints out the taxpayer number, filing status (using words, not single characters), taxable income, tax rate, and tax amount owed. (Use a **switch** statement on Filing Status, then a nested **if-else** to determine the tax rate.) Remember to use mnemonic variable names, to prompt the user for input, and to properly label and format the output (use a dollar sign, 2 digits after the decimal point, etc., on output).

Look on the back of this page!

Once you have 1-3 running correctly (and only ***after*** you have 1-3 running correctly!), add a “big” loop, which will allow your program to work for multiple taxpayers. This “big” loop will have as its body the code you wrote for 1-3, plus a bit more. Your program will allow the user to enter taxpayer data for multiple taxpayers. You’ll still print out information about each individual taxpayer. Your program will continue to run until she enters the value 0 (zero) for a taxpayer ID. (Inputting the number zero for the taxpayer ID will terminate the loop.) This might require you to slightly restructure your original code.

You will want to keep track of the following values:

- the **number of taxpayers** processed,
- the **highest tax amount,**
- the **taxpayer ID of the highest tax amount,**
- the **total amount of taxes paid,** and
- the **average tax amount.**

(**HINT**: Which of these can you determine *inside* the big loop and which of these must you wait until *after* the big loop is finished?) The program then prints a summary including the **number of taxpayers** processed, the **highest tax amount,** the **taxpayer ID of the highest tax amount,** the **total amount of taxes paid,** and the **average tax amount.** Of course, you’ll want to label and properly format these, too! ☺