

**Opeyemi Adesina, PhD**

Assistant Professor

Computer Information Systems

University of the Fraser Valley

Office: C2435, Abbotsford Campus**Tel:** (604) 504-7441 (ext: 4931)opeyemi.adesina@ufv.ca

Assignment 1 — Basic Java Constructs

COMP 155 : Object-Oriented Programming

(100 points)

When Due : June 10, 2021 – 23:59:00 (PDT) [Submission via Blackboard]

Brief Description

This assignment amounts to **10%** of the entire course grade. In particular, whatever you obtain as a score will be scaled to this value for final grade computation. You are required to work **ALONE**. No late submission will be permitted (see deadline above).

The goal of this assignment is to assess your knowledge and skills on Basic Java structures, while developing skills to map requirements (e.g. system model) to program code. You will find a grading scheme at the end of this document – to guide you on instructor's expectations while preparing your submission.

Program Requirements

In this assignment, we will be developing an extended version of the payroll deduction software (as developed in COMP 150, under my instructions).

1. Unlike the earlier version of the assignment, you will be developing system based upon seven provinces of Canada, plus the federal tax deduction. Employees are taxed at different levels - Federal and Provincial (i.e., employee's province of residence). Provinces of interest include: Alberta (AB - see Table 1), British Columbia (BC - see Table 2), Manitoba (MB - see Table 3), New Brunswick (NB - see Table 4), Nova Scotia (NS - see Table 5), Ontario (ON - see Table 6), and

Annual Taxable Income (range of incomes - \$)	Provincial Tax Rate (%)
0 .. 131,220.00	10.00
131,220.01 .. 157,464.00	12.00
157,464.01 .. 209,952.00	13.00
209,952.01 .. 314,928.00	14.00
314,928.01 .. (and above)	15.00

Table 1: 2021 Alberta's tax rates and income thresholds

Annual Taxable Income (range of incomes - \$)	Provincial Tax Rate (%)
0 .. 42,184.00	5.06
42,184.01 .. 84,369.00	7.70
84,369.01 .. 96,866.00	10.50
96,866.01 .. 117,623.00	12.29
117,623.01 .. 159,483.00	14.70
159,483.01 .. 222,420.00	16.80
222,420.01 .. (and above)	20.50

Table 2: 2021 British Columbia's tax rates and income thresholds

Annual Taxable Income (range of incomes - \$)	Provincial Tax Rate (%)
0 .. 33,723.00	10.80
33,723.01 .. 72,885.00	12.75
72,885.01 .. (and above)	17.40

Table 3: 2021 Manitoba's tax rates and income thresholds

Annual Taxable Income (range of incomes - \$)	Provincial Tax Rate (%)
0 .. 43,835.00	9.68
43,835.01 .. 87,671.00	14.82
87,671.01 .. 142,534.00	16.52
142,534.01 .. 162,383.00	17.84
162,383.01 .. (and above)	20.30

Table 4: 2021 New Brunswick's tax rates and income thresholds

Annual Taxable Income (range of incomes - \$)	Provincial Tax Rate (%)
0 .. 29,590.00	8.79
29,590.01 .. 59,180.00	14.95
59,180.01 .. 93,000.00	16.67
93,000.01 .. 150,000.00	17.50
150,000.01 .. (and above)	21.00

Table 5: 2021 Nova Scotia's tax rates and income thresholds

Annual Taxable Income (range of incomes - \$)	Provincial Tax Rate (%)
0 .. 44,740.00	5.05
44,740.01 .. 89,482.00	9.15
89,482.01 .. 150,000.00	11.16
150,000.01 .. 220,000.00	12.16
220,000.01 .. (and above)	13.16

Table 6: 2021 Ontario's tax rates and income thresholds

Annual Taxable Income (range of incomes - \$)	Provincial Tax Rate (%)
0 .. 45,677.00	10.50
45,677.01 .. 130,506.00	12.50
130,506.01 .. (and above)	13.16

Table 7: 2021 Saskatchewan's tax rates and income thresholds

Annual Taxable Income (range of incomes - \$)	Federal Tax Rate (%)
0 .. 49,020.00	15.00
49,020.01 .. 98,040.00	20.50
98,040.01 .. 151,978.00	26.00
151,978.01 .. 216,511.00	29.00
216,511.01 .. (and above)	33.00

Table 8: 2021 federal tax rates and income thresholds

Saskatchewan (SK - see Table 7). In table 8, you will find the tax rates for the federal level. Each table contains tax brackets and associated percentages.

- Other deductions like Canadian Pension Plan (CPP), Employment Insurance, and health premiums computed according to salary brackets. These include CPP (Canada Pension Plan) and EI (Employment Insurance). CPP is 5.25% of the gross income ([but to a maximum amount of \\$2,898.00](#)). EI is 1.58% of the gross income ([but to a maximum amount of \\$856.36](#)). In addition to the deductions above let us assume all employees (irrespective of their province of residence) are charged health premiums and that the premiums are based on the following rules:

- If the taxable income is less than or equal to \$22,000.00, the premium is \$0.00.
- If the taxable income is greater than \$22,000.00 and less than or equal to \$38,000.00, the premium is equal to the lesser of: (i) \$300.00 and (ii) 6% of (taxable income - \$22,000.00);
- If the taxable income is greater than \$38,000.00 and less than or equal to \$50,000.00, the premium is equal to the lesser of: (i) \$450.00 and (ii) 300 + 6% of (taxable income - \$38,000.00);
- If the taxable income is greater than \$50,000.00 and less than or equal to \$74,000.00, the premium is equal to the lesser of: (i) \$600.00 and (ii) 450 + 25% of (taxable income - \$50,000.00);
- If the taxable income is greater than \$74,000.00 and less than or equal to \$202,000.00, the premium is equal to the lesser of: (i) \$750.00 and (ii) 600 + 25% of (taxable income - \$74,000.00);
- If the taxable income is greater than \$202,000.00, the premium is equal to the lesser of: (i) \$900.00 and (ii) 750 + 25% of (taxable income - \$202,000.00).

- The model in figure 1 represents the internal structure of the system given to you. It is represented in a unified modelling notation with the goal of helping you understand the underlying structure and to help you navigate the code. **You are required to provide implementations for the following:**

20 points GovernmentTax.TaxUtil Class

- static Map<Integer, TaxCategory> alberta()
 - creates a map of categories, instantiates and add instances of each tax bracket in AB
 - returns a map of categories
- static Map<Integer, TaxCategory> britishColumbia()
 - creates a map of categories, instantiates and add instances of each tax bracket in BC
 - returns a map of categories
- static Map<Integer, TaxCategory> newBrunswick()
 - creates a map of categories, instantiates and add instances of each tax bracket in NB
 - returns a map of categories
- static Map<Integer, TaxCategory> novaScotia()
 - creates a map of categories, instantiates and add instances of each tax bracket in NS
 - returns a map of categories
- static Map<Integer, TaxCategory> ontario()
 - creates a map of categories, instantiates and add instances of each tax bracket in ON

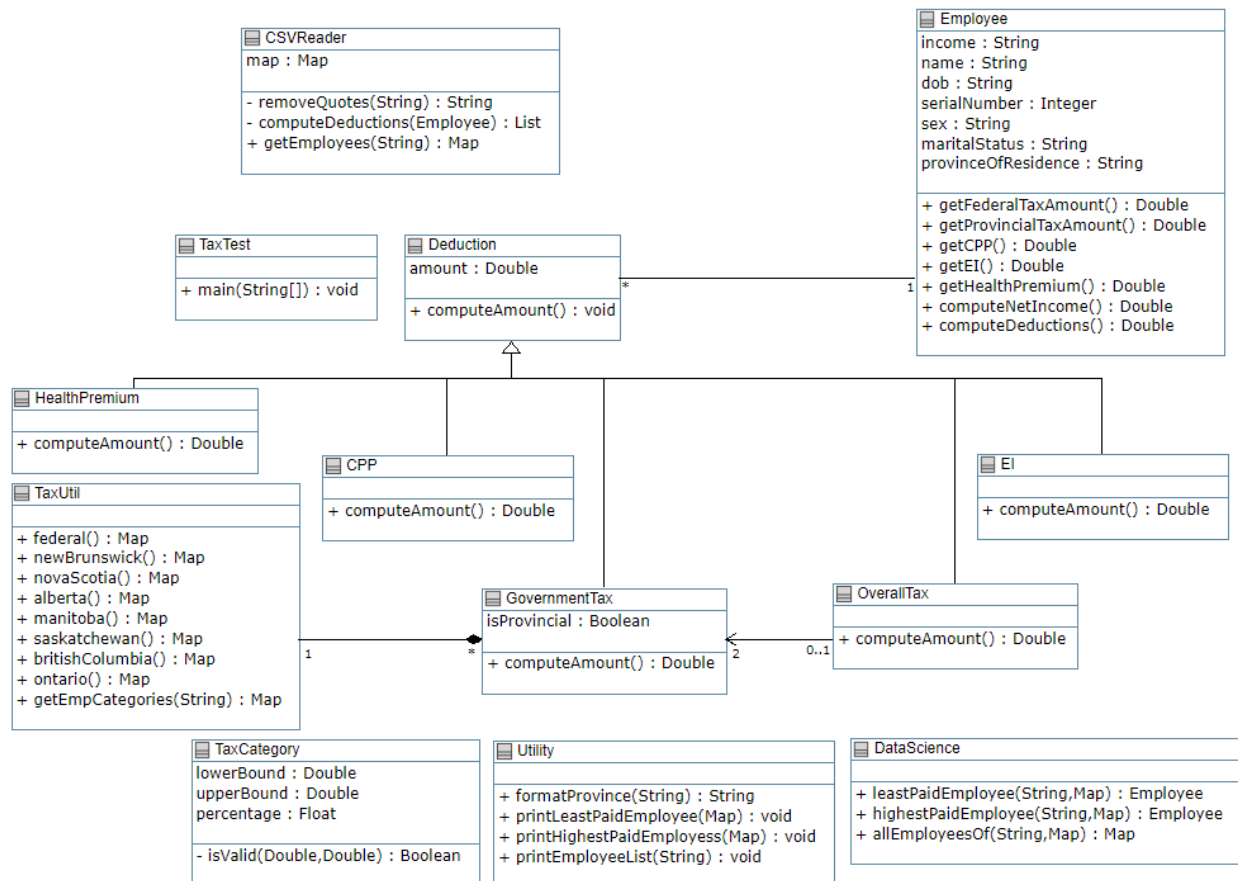


Figure 1: Canadian payroll system model

- returns a map of categories
- static `Map<Integer, TaxCategory> saskatchewan()`
 - creates a map of categories, instantiates and add instances of each tax bracket in SK
 - returns a map of categories
- static `Map<Integer, TaxCategory> federal()`
 - creates a map of categories, instantiates and add instances of each tax bracket at the federal level
 - returns a map of categories

20 points GovernmentTax Class (a template for instantiating provincial or federal tax object)

- double `computeAmount()`
 - calculates the corresponding taxes according to relevant categories.
 - returns the amount calculated

NB: For example, if an employee who is a resident of Ontario earns an income of \$93,230.00 – the taxes are computed below:

- * **Provincial Tax: \$6,771.54** = \$(2,259.37 + 4,093.89 + 418.28)
 - 1st category \$44,740.00 is tax at the rate of 5.05% \Rightarrow \$2,259.37
 - 2nd category (\$89,482.01 - \$44,740.01) is taxed at the rate of 9.15% \Rightarrow \$4,093.89
 - Remainder (i.e., \$93,230.00 - \$89,482.01) is taxed at the rate of 11.16% \Rightarrow \$418.28
- * **Federal Tax: \$16,442.72** = \$(7,280.25 + 9,162.47)
 - 1st category \$48,535.00 is tax at the rate of 15.0% \Rightarrow \$7,280.25

- Remainder (i.e., \$93,230.00 - \$48,535.01) is taxed at the rate of 20.5% \Rightarrow \$9,162.47

5 points EI Class

- double computeAmount()
 - calculates the employment insurance of an employee using the rules given above
 - returns the amount calculated

5 points CPP Class

- double computeAmount()
 - calculates Canadian pension plan of an employee using the rules given above
 - returns the amount calculated

10 points Health Premium Class

- double computeAmount()
 - calculates employee's health premium using the rules given above
 - returns the amount calculated

20 points DataScience Class

- Employee leastPaidEmployee(String province, Map \langle Integer, Employee \rangle map)
 - iterates over a list of employees for a given province (i.e. map) to obtain one of the employees with the least annual income
 - returns the employee found
- Employee highestPaidEmployee(String province, Map \langle Integer, Employee \rangle map)
 - iterates over a list of employees for a given province (i.e. map) to obtain one of the employees with the highest annual income
 - returns the employee found

For implementation purposes you will find statements of the form: "INSERT YOUR CODE HERE ..."
Apart from these places, do not modify any other part of the program unless you have consulted with the instructor. I have included a CSV file of employees upon which your program is being (to be) developed. An example output for a run is included in the given stub codes as well.

To compile and execute your code, please follow these chain of commands in the given sequence:

- `javac -d . *.java` (from the directory container the files - hard compile, compiling classes into their respective packages)
- `java TaxTest` (from the same directory - program execution)

```

C:\Users\adesino\Documents\UFV\COMP 155\Assignments\[firstName_LastName_StudentID]>javac -d . *.java
C:\Users\adesino\Documents\UFV\COMP 155\Assignments\[firstName_LastName_StudentID]>java TaxTest
..... Least Paid Employees .....
ID: 5863 Province: SK Gross Income: $13,704 Deductions: $4,430.5 (FTax: $2,055.6, PTax: $1,430.02, CPP: $719.46, EI: $216.52, HP: $0) Net Income: $9,273.5
ID: 5963 Province: ON Gross Income: $13,704 Deductions: $3,683.63 (FTax: $2,055.6, PTax: $692.05, CPP: $719.46, EI: $216.52, HP: $0) Net Income: $10,020.37
ID: 6063 Province: MB Gross Income: $13,704 Deductions: $4,471.61 (FTax: $2,055.6, PTax: $1,480.03, CPP: $719.46, EI: $216.52, HP: $0) Net Income: $9,232.39
ID: 3053 Province: NS Gross Income: $13,704 Deductions: $4,196.16 (FTax: $2,055.6, PTax: $1,204.58, CPP: $719.46, EI: $216.52, HP: $0) Net Income: $9,507.84
ID: 5563 Province: BC Gross Income: $13,704 Deductions: $3,685 (FTax: $2,055.6, PTax: $693.42, CPP: $719.46, EI: $216.52, HP: $0) Net Income: $10,019
ID: 9147 Province: AB Gross Income: $39,649 Deductions: $13,070.27 (FTax: $5,947.35, PTax: $3,964.9, CPP: $2,081.57, EI: $526.45, HP: $450) Net Income: $26,578.73
ID: 5019 Province: NB Gross Income: $13,704 Deductions: $4,318.13 (FTax: $2,055.6, PTax: $1,326.55, CPP: $719.46, EI: $216.52, HP: $0) Net Income: $9,385.87
..... Highest Paid Employees .....
ID: 5862 Province: SK Gross Income: $249,744 Deductions: $96,853.22 (FTax: $61,107.43, PTax: $31,091.43, CPP: $2,898, EI: $856.36, HP: $900) Net Income: $152,890.78
ID: 5962 Province: ON Gross Income: $249,744 Deductions: $91,295.17 (FTax: $61,107.43, PTax: $25,533.38, CPP: $2,898, EI: $856.36, HP: $900) Net Income: $158,448.83
ID: 6062 Province: MB Gross Income: $249,744 Deductions: $105,170.49 (FTax: $61,107.43, PTax: $39,408.7, CPP: $2,898, EI: $856.36, HP: $900) Net Income: $144,573.51
ID: 3052 Province: NS Gross Income: $249,744 Deductions: $109,345.48 (FTax: $61,107.43, PTax: $43,592.69, CPP: $2,898, EI: $856.36, HP: $900) Net Income: $140,398.52
ID: 5562 Province: BC Gross Income: $249,744 Deductions: $97,336.01 (FTax: $61,107.43, PTax: $31,574.22, CPP: $2,898, EI: $856.36, HP: $900) Net Income: $152,407.99
ID: 9136 Province: AB Gross Income: $231,194 Deductions: $85,708.89 (FTax: $54,985.93, PTax: $26,068.6, CPP: $2,898, EI: $856.36, HP: $900) Net Income: $145,485.11
ID: 5018 Province: NB Gross Income: $249,744 Deductions: $106,840.22 (FTax: $61,107.43, PTax: $41,078.43, CPP: $2,898, EI: $856.36, HP: $900) Net Income: $142,903.78
Time taken : 218 (ms)
Test 0 (Q1) passed. [Alloted point(s): 2.80]
Test 1 (Q2) passed. [Alloted point(s): 4.0]
Test 2 (Q3) passed. [Alloted point(s): 2.80]
Test 3 (Q4) passed. [Alloted point(s): 2.80]
Test 4 (Q5) passed. [Alloted point(s): 2.80]
Test 5 (Q6) passed. [Alloted point(s): 1.71]
Test 6 (Q7) passed. [Alloted point(s): 2.80]
Test 7 (Q8) passed. [Alloted point(s): 2.5]
Test 8 (Q9) passed. [Alloted point(s): 2.5]
Test 9 (Q10) passed. [Alloted point(s): 2.5]
Test 10 (Q11) passed. [Alloted point(s): 2.5]
Test 11 (Q12) passed. [Alloted point(s): 2.5]
Test 12 (Q13) passed. [Alloted point(s): 2.5]
Test 13 (Q14) passed. [Alloted point(s): 2.5]
Test 14 (Q15) passed. [Alloted point(s): 2.5]
Test 15 (Q16) passed. [Alloted point(s): 5.0]
Test 16 (Q17) passed. [Alloted point(s): 5.0]
Test 17 (Q18) passed. [Alloted point(s): 10.0]
Test 18 (Q19) passed. [Alloted point(s): 10.0]
Test 19 (Q20) passed. [Alloted point(s): 10.0]
All tests are successful.
Total points earned in this lab: 80.0 points

```

Figure 2: Expected Output

Grading Scheme

The following scheme will be used to grade your submission. Therefore, you may also use it as a guide in preparing your deliverable.

Grade Item	Weight
A syntactically and semantically correct program.	80
Program efficiency. That is, using efficient but correct control structures.	10
A program with detailed program documentation and uses sensible variable names. Your program's file name and other files should be zipped and named in the following format - [first-Name.lastName.studentID]	10
Total	100