GMU Fall 2020 – CS 211 – Project 2

Due Date: Sunday, October 11th, 11:59pm

Description

The purpose of this assignment is to practice object oriented programming principles with Java. Your task is to implement a set of classes that can be used toward the building of an application that does taxation calculations like the ones you encounter during the filing of the federal tax form 1040. Keep in mind that, for training reasons, the logic you're asked to implement here does not necessarily conform to the IRS rules.

Guidelines

- Honor Code: The project is **individual work** of each student, you're not allowed to collaborate in any form. Copying code from other sources (colleagues, websites, etc.) is a serious violation of the University's Honor Code. Your code will be examined for similarities with other sources.
- Validation: Make sure all method inputs are properly validated (e.g. *income* can't be negative, *family members* can't be zero, etc.)
- Documentation: Comments are required. You may use JavaDoc-style comments, but for this assignment they are not required to be JavaDoc-style. You should insert comments in all lines that it's not obvious what they do.
- Visibility: Class fields should not in general be visible/accessible to other classes. Create getter and setter methods for any attributes that you want to exchange with other objects.
- Addons: You may add your own helper methods or data fields. Additional methods can have any visibility but additional attributes must be private only. You may not import any package.

Submission

Submission instructions are as follows:

1. Let xxx be your lab section number, and let yyyyyyyy be your GMU userid. Create the directory xxx_yyyyyyyy_P2/
2. Place your files in the directory you've just created.
3. Create the file ID.txt in the format shown below, containing your name, userid, G#, lecture section and lab section, and add it to the directory.
    1. Full Name: George Mason
    2. userID: gmason
    3. G#: 01234567
    4. Lecture section: 003
    5. Lab section: 213
4. Compress the folder and its contents into a .zip file, and upload the file to Blackboard.

Assumptions

You may assume that:

- A child can have an income but this income cannot be a wage and, therefore, it's not subject to social security and medicare taxes or tax withholding.
- A married person cannot file as "single"
- A single-parent family will file as "single"
- We're not considering cases of non-children dependents.
- A couple has the option to file either "jointly" or "separately". When a couple is filing "separately" only one spouse is added to the family's tax return.

Grading

- If your program doesn't compile you don't get any credit.
- If your program runs but it crashes due to runtime errors, the maximum you'll get is 30 instead of 100 points.
- 10 points for comments, structure and readability of the code
- 90 points for correct logic and implementation

Tasks

You must implement the following classes based on the specification provided. Be reminded that this is the minimum implementation required, you're free to add more methods and attributes as you see fit. However, keep in mind that although additional methods can have any visibility, additional attributes must be private only.

Person [15pts]

id	private unique positive number (the type is your call) that starts at 1 and increments by 1 in every object instantiation
name	private string (cannot include any character other than a-z or A-Z or space)
birthday	private string (in YYYY/MM/DD format)
ssn	private string (nine-digit SSN in xxx-xx-xxxx format)
income	gross income stored as private non negative float
Person()	No parameters. Initializes the id
setName(string)	Returns false if invalid input or true otherwise
setBirthday(string)	Returns false if invalid input or true otherwise (do not make a complete validation, just make sure the substrings between / include digits only)
setSSN(string)	Returns false if invalid input or true otherwise
setIncome(float)	Returns false if invalid input or true otherwise
getIncome()	getter method for income

getId()	getter method for id
toString()	Returns the person's name followed by masked SSN and birthday in the following format: George Mason xxx-xx-1234 YYYY/**/**
deduction(Family)	Empty placeholder to be overridden by child classes. Returns 0.0

Adult <inherits from Person> [15pts]

employer	private string
Adult(name, birthday, ssn, income, employer)	Constructor with five parameters. It uses parent methods (if available) to set the respective attributes.
toString()	Overrides parent method. Returns the person's name followed by masked SSN, masked birthday, and gross income, in the following format: George Mason xxx-xx-1234 YYYY/**/** $123456.78
adjustedIncome()	Returns the adjusted income of the employed adult. Adjusted income is what remains in the gross income after subtracting the social security and medicare taxes. Both social security and medicare taxes are split equally between employer and employee. Also, keep in mind that there is an income limit that the social security tax applies to. These three parameters might change over time, so don't hard code them but use the Taxation object to store and retrieve them.
taxWithheld()	Returns the amount of tax that was withheld from an Adult's paychecks. Tax withholding is done at a progressive rate on the gross income. It's 10% for the first $50K, 15% on the next $100K, and 20% thereafter.
deduction(Family)	Overrides parent method. It returns the amount of income that is exempted from taxation. First of all, it can't be higher than the adjusted income. The base exemption is defined in Taxation. If it's a single parent family, the base exemption is doubled. If the adjusted income is above $100,000 the exemption is reduced by 0.5% per whole thousand dollars of adjusted income above 100K. Regardless of the income level, though, this reduction cannot exceed 30%.
getEmployer()	getter method for employer

Child <inherits from Person> [10pts]

school	private string
tuition	amount of tuition paid per year (private)
Child(name, birthday, ssn, income, school, tuition)	Constructor with six parameters. It uses parent methods (if available) to set the respective attributes.
toString()	Overrides parent method. Returns the child's name followed by masked SSN, masked birthday, and the school, in the following format: George Mason xxx-xx-1234 YYYY/**/** Fairfax High School
getTuition()	getter method for tuition
deduction(Family)	Overrides parent method. It returns the amount of income that is exempted from taxation. First of all, it can't be higher than the income. Base exemption for each child is defined in Taxation. If the family has more than two children, the child exemption is reduced by 5% for each additional child (i.e. for 3 kids -5%, for 4 kids

-10%, and so forth). Regardless of the number of children, though, the child's exemption cannot be reduced by more than 50%.

Family [20pts]

numMembers	total number of family members (private)
filingStatus	1=single, 2=married filing jointly, 3=married filing separately (private)
Person[]	An array of family members
Family(members, filingStatus)	Constructor initializes the two attributes and allocates memory for the array of members.
addMember(Person)	Adds a family member to the array
getNumAdults()	Returns the number of family members that are adults/parents
getNumChildren()	Returns the number of family members that are children
getFilingStatus()	getter method for filingStatus
getTaxableIncome()	Returns the family's total taxable income (i.e. income left after subtracting the deductions from the adjusted income).
taxCredit()	Returns the amount of tax credit that a family is eligible to receive. A family is eligible for a tax credit if its taxable income is in the low 50% of the median income per capita. The most recent median income per capita in the United States was $31,099 (see Taxation). The credit is $30 per each whole thousand dollars of taxable income. Each child is eligible for an additional credit which is equal to its tuition or $1000, whichever is lower. If parents are filing separately they can each claim half of it only. Maximum credit per family is $2,000 or the amount of pre-credit tax, whichever is lower.
calculateTax()	Returns the amount of tax that a family either owes or is to be refunded with. This method must call the other methods and then use their results. It must not repeat any of the calculations that have already been done in the other functions. First, calculate the taxable income based on the exemptions that a family qualifies for. Then, find the maximum tax bracket of the taxable income. Then, sum the amounts of tax that correspond to each bracket up to the max bracket. Once you calculate the total tax, find if the family is eligible for a tax credit and subtract it from the total tax. And finally, subtract the tax that was already withheld during payroll.

Taxation <all members are static> [10pts]

socialSecurityRate	public constant float value (default is 12.4%)
socialSecurityIncomeLimit	public constant float value (default is $137,700)
medicareRate	public constant float value (default is 2.9%)
adultBaseExemption	public constant float value (default is $3,000)
childBaseExemption	public constant float value (default is $2,000)
medianIncomePerCapita	public constant float value (default is $31,099)
incomeBrackets	You can store this data any way you want but it has to be private. Other objects will use bracketIncome() to query this table.

bracket #	single	married (jointly)	married (separately)

1	$0 to $10,000	$0 to $20,000	$0 to $12,000
2	$10,000.01 to $40,000	$20,000.01 to $70,000	$12,000.01 to $44,000
3	$40,000.01 to $80,000	$70,000.01 to $160,000	$44,000.01 to $88,000
4	$80,000.01 to $160,000	$160,000.01 to $310,000	$88,000.01 to $170,000
5	$160,000.01 or more	$310,000.01 or more	$170,000.01 or more

taxRates — You can store this data any way you want but it has to be private. Other objects will use bracketTaxRate() to query this table.

bracket #	**single**	**married (jointly)**	**married (separately)**
1	10%	10%	10%
2	12%	12%	12%
3	22%	23%	24%
4	24%	25%	26%
5	32%	33%	35%

getNumTaxBrackets() — Returns the number of tax brackets. You need this because the above tables are private.

maxIncomeTaxBracket(Family) — returns the maximum tax bracket that a family's income is in

bracketIncome(Family, int b) — Returns the portion of a family's taxable income that falls within bracket b

bracketTaxRate(int b, int f) — Returns the tax rate that corresponds to bracket b and filing status f

TaxYear **[15pts]**

TaxYear(int max) — The constructor takes one parameter, the maximum number of returns that can be filed for this year

taxFiling(Family) — Files a family's tax return for the year and validates the submitted data. If there is any kind of error (e.g. filing status doesn't agree with family members, a family doesn't have a parent, etc.), it doesn't accept the filing and returns false. If the data is valid it adds the Family to a local storage (the datatype is your call) and returns true. Families are stored in consecutive locations in the order they're added.

taxWithheld() — Returns the total tax that was withheld from all families' paychecks up to the moment of invoking this method (i.e. it may be called before tax filing for the

	year is complete)
taxOwed()	Returns the total tax that is owed by all families (i.e. based on their taxable income only), at the moment of invoking this method (i.e. it may be called before tax filing for the year is complete)
taxDue()	Returns the total tax that was due/returned the moment of invoking this method (i.e. it may be called before tax filing for the year is complete)
taxCredits()	Returns the total tax credits that were given to families the moment of invoking this method (i.e. it may be called before tax filing for the year is complete)
numberOfReturnsFiled()	Returns the number of tax returns that were filed
numberOfPersonsFiled()	Returns the total number of persons that are included in the tax returns that were filed.
getTaxReturn(int index)	Returns the Family that is stored at location index of the local storage.

Analytics **[15pts]**

povertyThreshold	Default value for 2020 is $26200 per family according to Census Bureau
Analytics(TaxYear)	Constructs an object that will be used for running various statistics on a certain tax year
setPovertyThreshold(float)	Setter method for povertyThreshold
averageFamilyIncome()	Returns the average taxable income per family for the given year
averagePersonIncome()	Returns the average taxable income per person for the given year
maxFamilyIncome()	Returns the maximum family taxable income for the given year
maxPersonIncome()	Returns the maximum personal taxable income for the given year
familiesBelowPovertyLimit()	Returns the number of families that have a taxable income below the poverty threshold
familyRank(Family)	Returns the rank of a family's taxable income in a certain tax year. Assume no other family has the exact same income. The rank starts at 1.