CEGEP VANIER COLLEGE
Department of Computer Science

420-101-VA PROGRAMMING 1 sections 1 & 2

Assignment_03: Due November 3rd , 2023, at 11:59pm Prof. Nagat Drawel

Overview:

The purpose of this assignment is to learn:

1. 1-D array
2. Java static methods.
3. Method overloading
4. Passing Arguments to methods (Pass-by-value and Pass-by-reference)
5. Javadoc

Notes:

1. Please make sure to follow the programming standards (e.g., naming conventions, comments, @author); Keep your code clean (Indentation, Spaces, etc.,) not doing so will result in loss of marks.
2. Please demonstrate your working programs to your teacher. Failing to explain your own code will be heavily panelized.
3. Please submit one .java file for each task (3 java files in total, Omnivox accepts multi-submission). DO NOT ZIP YOUR FILES.
4. All methods must be documented by writing comments that appear just before the method's definition.

Task 01

Create a java project **SalaryCalculator** which contains the following methods:

1. **calculateYearlySalary (double hourlyWage)** method that takes in the user's hourly wage as a parameter and calculates their yearly salary, based on a working week of 40 hours and a working year of 50 weeks then returns the annual salary.
1. **calculateTax (double salary, int percentage)** method that takes in the user's annual salary returned from ***calculateYearlySalary*** method and calculates the taxes owed based on a specific tax bracket, which is given as a percentage (0% -100%) in a parameter. The taxes owed should be returned as a double.
2. **calculateSavings (double salary, int currAge, int retirementAge, int percent)** that takes in the user's annual salary after subtracting the taxes owed (returned from the method **calculateTax)**, current age, desired retirement age, and percent saved each year as

parameters. It will calculate how much the user will have saved by the time they retire. The return is double value.

Demonstrate each of the methods in the main method with appropriate way of your choice.

Task 02

Create a Java project **SimpleCalculator** that will call the following three overloaded methods to calculate and print the required results:

1. **calcResult(double number1, double number2)**: calculates addition, with two numbers: **number1** and **number2**
2. **calcResult (double number1, double number2, char oper)**: calculates +, -, *, or / based on the **oper**, with two numbers: **number1** and **number2**. **(Assume the user will only input +, -, *, or /)**
3. **calcResult (String formula)**: takes a formula (**number1 oper number2**) looks like: "31 * 2" or "3.14 + 7.1", (there is a space between **number1** and **oper**, and a space between **oper** and **number2**), you have to extract the **number1**, the **oper**, the **number2** from the string, and then calculate the result.

Demonstrate each of the methods in the main method with appropriate way of your choice.

Task 03

1. Write a class named `IntArrayTools` that includes the following static methods:
    a. `initializeArray:` This method should accept a 1-D array as its argument and initialize the array so that the first half of the array elements are equal to the square of the index variable, and the second half elements are equal to three times the index variable.
    b. `getAverage:` This method should accept a 1-D array as its argument and return the average of the odd values in the array.
    c. `smallestIndex:` This method should accept a 1-D array as its argument and returns the number of the occurrence of the smallest element in the array.
    d. `switchElement:` This method should accept a 1-D array as its argument and two indexes `idx1` and `idx2`. It should replace the element at `idx1` with the `idex2` element then return the array.
    e. `arraySort:` This method should accept a 1-D array then sort the array in ascending order. Return the sorted array.

Demonstrate each of the methods in the program with an array that is of even size. Use any primitive data type of your choice.