Assignment 2: More Tinkering with Java

Assignment for COSC 1200 – Object Oriented Programming I

Prior to attempting this problem, you should have done the following:
1. Viewed the content, including assigned readings for weeks 1, 2, 3, 4, 5, 6.
2. Viewed the lectures and practiced with the demo code for weeks 1, 2, 3, and 4

General Requirements

1. This assignment is to be completed individually.
2. Analyze the problem, it is recommended (but not required, will not be graded) to develop a plan your solution using a flowchart or pseudo code.
3. Submit your solution to the appropriate assignment folder on DC Connect **by the due date provided**.
4. Your instructor will assign a grade and post feedback on your submission to DC Connect.
5. Note: Each class needs comments at the top with at minimum your name(s), the date (date created or date due), a description of what the class does. In addition, it is expected that you provide a sufficient amount of commenting in your code.

Requirements

Complete and submit the program source code that satisfies the following requirements:

1. Create a new class named ScoreArray.java. This class will have a main() method that does the following:
    a. Declares and initializes an array of 8 (8) integers. Name the array appropriately
    b. User a decision structure of your choice to ensure input is validated.
        i. Only accept numeric data in the form of an integer
        ii. The acceptable range is 0 to 100 i inclusive.
        iii. If user input is not valid, display an appropriate custom error message and re-prompt the user for valid data.
        iv. Hint: use a loop. No processing should occur if data is invalid.
    c. Declare numeric variables named averageScore, highestScore, lowestScore that is initially set to zero (0). These will hold the calculations for averages, highest, and lowest scores.
    d. Create a for loop that gets the length of the score array as the condition for continued looping. This for loop should display the score number and its value (e.g. “Score 1 = 96”).
    e. As well, each loop through should add the score to the average variable.
    f. You also need to determine and display the highest and lowest score in the array.
    g. After the loop finishes executing, display the average score, and highest score, and the lowest score. Note: do not use the literal value (i.e. the number 8) as the

denominator in the average calculation, best practice is to determine the length of the array.

2. Create a new class named PlayingWithStrings.java. This class will have a main() method that does the following:
    a. Instantiates and initializes a String object named lastName (should be initialize to YOUR LAST NAME)
    b. Display in a sentence the length of your last name (using the appropriate String method)
    c. Declare an integer named thirdLastLetter that is equal to 3 less than the length of your last name.
    d. Use an appropriate String method to display the third-to-last character of your last name (in a complete sentence).
    e. Use an appropriate String method to display your last name all-in upper-case letters.
    f. Use an appropriate String method to display your last name all-in lower-case letters.
    g. Create an if statement that compares your last name to the string “Short”, and displays whether or not you have the same last name as your professor. Add to that if statement an else if that compares your last name to the string “Patel”, and displays whether or not you have the same last name as a few students in our class.
    h. Create an if statement that determines whether your last name contains the character sequence “ne”. If your last name contains “ne” a statement should state that your last name (display it) contains the character sequence “ne”, or it should say that it does not. e.g. Display for the instructor will be:
        The last name Short does not contain “ne”
    i. Please take notice of the double quotes.
    Note: Google the Java API to find which method checks for character sequences in a String object.

3. Create a new class named FunWithMath.java. For this program you will be required to use the Math class (found in the java.lang package) and the DecimalFormat class and the NumberFormat.getCurrencyInstance() method (these are found in the java.text package and will need to be imported). Inside the main() method, do the following:
    a. This program will calculate the present value of a future amount, using the compound interest formula:

presentValue = futureValue/(1 + annualInterestRate)yearsInFuture

b. Declare primitives (and initialize them) for the following (choose appropriate data types) :
    i. currentValue = 0
    ii. futureValue = 1000000
    iii. annualInterestRate = this should be a random number between 0% and 10% (use the Math.random() method)
    iv. yearsInTheFuture = 20
c. Your program should calculate the present value of the one million dollars.
d. Also, instantiate two (2) different (appropriately named) DecimalFormat objects, one that will display the dollar amounts appropriately, and one that will display interest rates to 3 decimal places (a leading zero if the interest rate is less than 1%).

4. Create a new class named Temp.java. This class will have a main() method that will:
    a. Prompt the user for and accept user input a temperature in the Fahrenheit temperature scale.
    b. User a decision structure of your choice to ensure input is validated.
        i. Only accept numeric data in the form of a real number
        ii. The acceptable range is -130.0 F and 135.0 F inclusive.
        iii. If user input is not valid, display an appropriate custom error message and re-prompt the user for valid data.
        iv. Hint: use a loop. No processing should occur if data is invalid.
    c. Once input has been validated, pass the value to a function you need to create named TempConversion() that will accept a double or float data type (your choice). This function will calculate the temperature conversion and return the calculated value.
    d. Display the output that includes the original user input and the converted Celsius value.

Note: please only submit your .java files to DC Connect.

Style Guide Notes

To be eligible for full marks on this or any assignment in this course your application must conform to the requirements as outlined above as well as the Google style guide as well as appropriate and complete program documentation.