

420-101-VA

Teacher: Jaina Sheth

Programming 1

Assignment - 04

Vanier College

April 27, 2021

Deadline : 11th May, 2021

Time : 11:59 PM

Marks: 60 Points

Read the instructions carefully.

1 General Instructions

- This work is an **individual** work and should satisfy expectation of originality.
- Any form of plagiarism will not be tolerated.
- You must submit a single **.zip file** for this assignment on Lea under **420-101-VA Assignment-04**.
- Place all your Java programs (.java files) directly in a folder with a name **Lastname_Firstname_StudentID** (e.g., Sheth_Jaina_2115533). Compress (zip) the folder and submit a **.zip** file only. There shouldn't be any **package** folder or **src** folder inside the zipped folder.
- You must not submit files with extensions such as a **.rar**, **.tz**, **.7z**, **.java**, **.txt**, etc.
- If you submit a file other than **.zip** file, your submission will not be considered and you will be graded a straight 0. Also, if you submit multiple files, only the last submitted work will be considered.

2 Grading

You will be graded out of 60 points according to the following distribution:

- [10 points] 5 points for each of the tasks 1 and 2.
- [20 points] 10 points for each of the tasks 3 and 4.
- [30 points] 15 points for each of the task 5 and 6.

3 Tasks

Write a Java program for each of the following. Design a generalized solution which can work for any valid values of variables.

1. Create a method which calculate and return the value of simple interest for user entered **Principal amount**, **Rate of interest per annum**, and **Number of years**. Display the result on the screen along with principal amount, rate of interest per annum, and number of years.
2. Consider an array of 10 whole numbers and initialize it with the numbers of your choice. Then, ask the user to enter a whole number. Check if the array you created contains the user entered number or not and display the result on the screen as either “**The array contains the user entered number**” or “**The array does not contain the user entered number.**”
3. Consider an array of 5 whole numbers named **userArray**. Ask the user to enter values of array elements, store them inside **userArray** and display the elements of **userArray** on the screen.

Now, reverse the **userArray** and store the reversed values into another array called **revArray**. Display the elements of **revArray** on the screen.

4. Ask the user to enter a year between **1800** to **2200**. Check if the user entered value is in the given range or not. If in the range, proceed with the program; otherwise keep on asking the user to enter a year until user enters a valid value. Create a method to check if the user entered year is a leap year or not and display the result on the screen.
5. Create a **gradebook** class. Student IDs are going to be the objects of this class. The objects of this class should have the states: **maths**, **science**, **language**. The objects of this class should have the behaviours: **calculateTotal()**, **calculatePercentage()**, **finalGrade()**.
 - Create 3 different objects of the class **gradebook**: S01, S02 and S03.
 - **maths**, **science**, and **language** should accept grades for a student for subjects Mathematics, Science and Language respectively.
 - Add grades for the students while creating objects using parameterized constructor. The values you enter should be between **0 to 100**.
 - The method **calculateTotal()** should return total of grades of all the subjects for a student.
 - The method **calculatePercentage()** should return a student's percentage out of 100 depending on the total of the grades.
 - The method **finalGrade()** should return **PASS** or **FAIL** for a student depending on Percentage of the student. If the percentage for a student is below 60, return **FAIL**; otherwise return **PASS**.
 - Display the total, percentage and final grade on the screen for each student.

6. Create a method to calculate sum of the series of numbers considered according to the following expression for user decided range: (Please refer to the given examples before implementing.)

$$n/(n+2)$$

- Ask the user to enter the minimum and maximum number of the range. Note that these should be whole numbers.
- The **method should accept these minimum and maximum numbers** of the range entered by the user, as its arguments. The **method should return sum** of the series of numbers according to above expression for the user decided range.
- **Display the series of numbers** according to the expression and for user entered range along with **the sum of the series** on the screen.
- Displayed series should have the numbers as whole numbers without a decimal point as shown below in the example. Whereas, the sum of the numbers should be a floating point number.

Example-1: For example, if the user entered minimum number is 1 and user entered maximum number is 99; the program should display the following series and sum:

1/3
3/5
5/7
7/9
.
.
.
95/97
97/99

Sum : 45.124450303050196

Example-2:For example, if the user entered minimum number is -5 and user entered maximum number is 5; the program should display the following series and sum:

-5/-3
-3/-1
-1/1
1/3
3/5

Sum : 4.6

(Your program should work for any valid minimum and maximum values of the range.)