

## PRACTICE SHEET WEEK - 2

### Topics:

- Class creation in VS Code.
- Deploying code to the salesforce org.
- Methods Inside class.
- Variable declaration and definition.
- Using String class and Date class methods. Link to [String Class](#) methods.
- Classes, Class Objects/Instantiation.
- Parameterized methods
- If-else, switch

### IMPORTANT NOTE:

**USE WEEKEND DEV PRACTICE ORG AND THE PROJECT YOU CREATED IN WEEK 1.**

### Practice Questions:

1. Create a class with the name '**ValidateAddress**'.  
Create a method inside the ValidateAddress class with the name '**validateHomeAddress**'. Inside **validateHomeAddress** method, define different variables to store the following information:  
[Please provide default values in the variables]
  - a. House Name
  - b. Street Name
  - c. City Name
  - d. State Name
  - e. State ISO Code
  - f. Country Name
  - g. Country ISO Code
  - h. Now include the debug messages for the following requirements:
    - i. The number of characters in the street name.
    - ii. Check if the street name contains a house name in it or not.
    - iii. Find if the street name contains the state iso code in it
    - iv. Convert the state iso code, country iso code, country name to uppercase and city name, and state name to lowercase.
    - v. Concatenate the house name, street name, City Name, State Name, and Country Name and store it in a variable.

Once you are done with the above steps, create an Apex script to execute the validateHomeAddress method of ValidateAddress class anonymously.

2. Create a class named as **Planets**. *[OPTIONAL]*

Each planet has a following details:

- name,
- planet number in the solar system,
- number of moons the planet has.

Declare the class variables accordingly.

- a. Create a new anonymous apex file.
    - i. Create objects for Earth
    - ii. Create an object for your favorite non-earth planet.  
And print the object values in the debug message.
- 

3. Create a class named **Students**. Each student has a **first name**, **last name**, **class year**, and **major/stream** (like Physics, Electrical).

Declare the class variables accordingly.

Create a method with the name as **generateReport**.

This method is going to show a debug message in the following format and return the value of major/stream.

Debug message format '**Report Generated for [first name] [last name] of [class year]**'.

Now create at least two instances of the class to store the information of two different students and call the generateReport method for each.

Try to print the value returned from the generateReport method as well.

---

4. Create a class named **OperationsHelper**. *[Optional]*

Now create the following methods in the class as per the given requirement:

- a. Method Name – **largestOutOfTwoNumbers**  
Description - It should have two parameters that can accept two decimal numbers and should return the largest number.
- b. Method Name – **smallestOutOfThreeNumbers**  
Description - It should have three parameters that can accept three decimal numbers and should return the smallest number.
- c. Method Name - **sumofPositiveIntegers**  
Description - It should return the summation of first n numbers. n is the input parameter

## PRACTICE SHEET WEEK - 2

of type Integer. if n is negative then show this message- "We are not calculating the sum for negative numbers" and return null value.

---

5. Create a class named **Calculator** class,  
create a method and name it as **dynamicCalulationMethod**.  
Should it be a static method or a non-static method? Discuss among the team members.  
**dynamicCalulationMethod** should have three parameters.  
First two parameters are for passing the decimal values and  
The third one is for passing the operation that the user wants to do.

For Ex: Calling dynamicCalulationMethod with the following parameters should give the output shown below.

- a. Input → 10, 67.5, ADDITION      Output → 77.5
- b. Input → 90, 60, SUBTRACTION      Output → 30
- c. Input → 8, 2, DIVISION      Output → 4
- d. Input → 6,4, MULTIPLICATION      Output → 24
- e. dynamicCalulationMethod should return the value after doing the calculation.

---

**END**

---