PRACTICE SHEET WEEK - 3

Topics:

1. Classes, Class Objects/Instantiation.
2. Parameterized class methods and with the return type.
3. Static, non-static in brief
4. List, Set, Map and its methods.

IMPORTANT NOTE:

USE Practice ORG AND Practice SFDX Project for the exercise.

DO NOT USE DAILY SESSION ORG.

Practice Questions:

1. **List Creation and List Class Methods**

   In a new anonymous apex file, create a List named as **coloursList**, store at least 5 color Names in it, and print it.

   a. Print the size of the above-created List.
   b. Check if the coloursList contains ‘Red’ color in it or not.
   c. Retrieve the last element of the List and print it in the upper case.
   d. Make the fourth element in the List as Upper case.
   e. Create a different List named colourCodesList in which we need to store some color codes. For example: ‘#FF0000’ is code for red color.
   f. Compare if the two lists colourCodesList and coloursList are equal or not.
   g. Merge the two Lists (colourCodesList, coloursList) and print the newly created list.

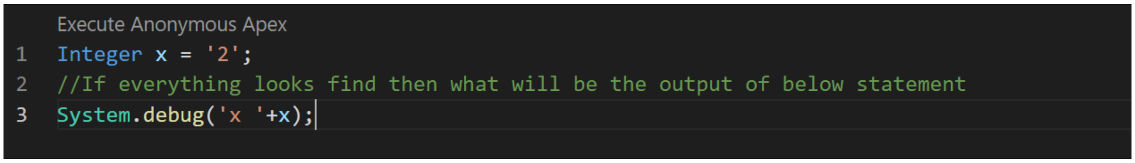
2. In a new anonymous apex file, create a Set named as **fruitsSet**, store at least 4 fruits name in it.

   a. Print fruits name from **fruitsSet** one by one.
   b. Print the size of the **fruitsSet**.
   c. Add a fruit name “Apricot” to the **fruitsSet** and print the **fruitsSet.**
   d. Check if the **fruitsSet** contains the fruit ‘Apricot’
   e. Check if the **fruitsSet** is empty?

PRACTICE SHEET WEEK - 3

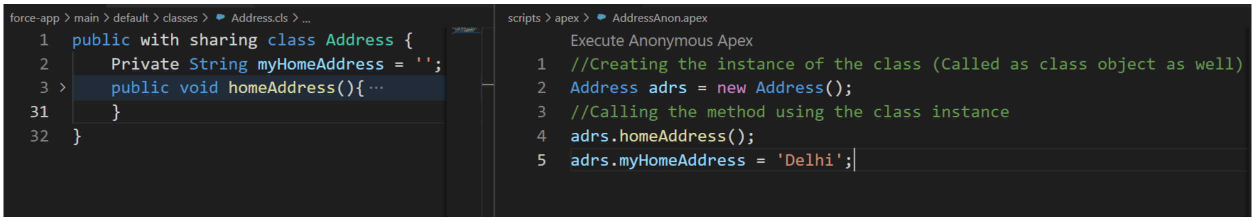
3. Find the problem in the below code snippets. Discuss with your group and mentors and just write notes of all the problems.

3.1



```
Integer x = '2';
//If everything looks find then what will be the output of below statement
System.debug('x '+x);
```

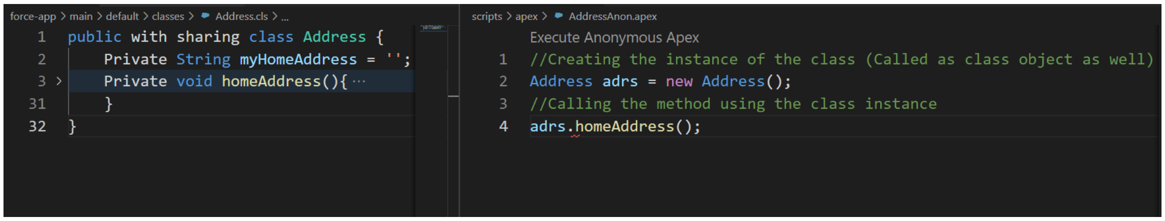
3.2


force-app > main > default > classes > Address.cls > ...
scripts > apex > AddressAnon.apex
Execute Anonymous Apex

```
public with sharing class Address {
    Private String myHomeAddress = '';
    public void homeAddress(){ ···
    }
}
```

```
//Creating the instance of the class (Called as class object as well)
Address adrs = new Address();
//Calling the method using the class instance
adrs.homeAddress();
adrs.myHomeAddress = 'Delhi';
```

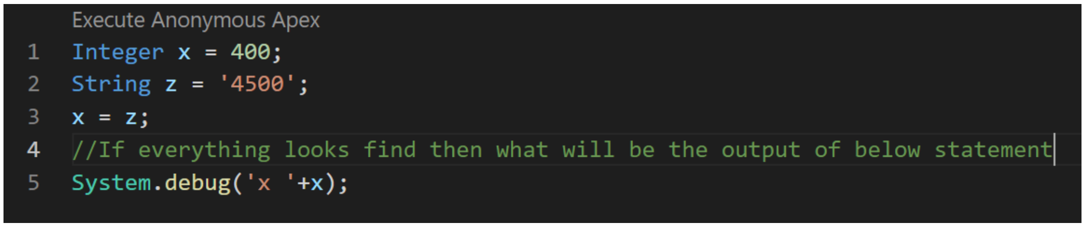
3.3


force-app > main > default > classes > Address.cls > ...
scripts > apex > AddressAnon.apex
Execute Anonymous Apex

```
public with sharing class Address {
    Private String myHomeAddress = '';
    Private void homeAddress(){ ···
    }
}
```

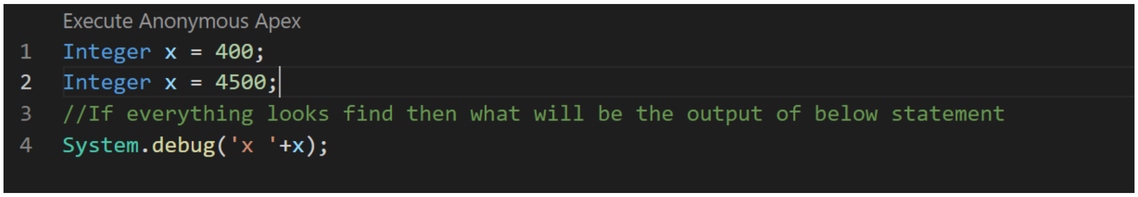
```
//Creating the instance of the class (Called as class object as well)
Address adrs = new Address();
//Calling the method using the class instance
adrs.homeAddress();
```

3.4



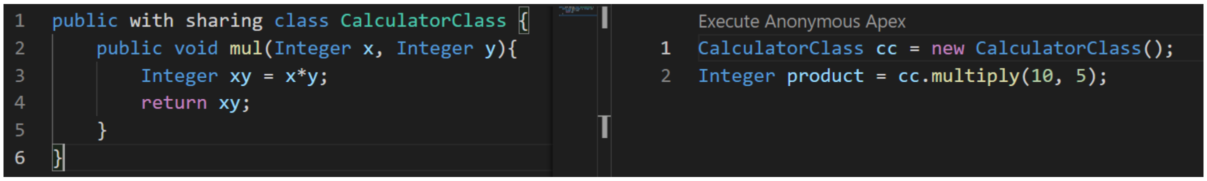
```
Integer x = 400;
String z = '4500';
x = z;
//If everything looks find then what will be the output of below statement
System.debug('x '+x);
```

3.5



```
Integer x = 400;
Integer x = 4500;
//If everything looks find then what will be the output of below statement
System.debug('x '+x);
```

3.6



```
public with sharing class CalculatorClass {
    public void mul(Integer x, Integer y){
        Integer xy = x*y;
        return xy;
    }
}
```

```
CalculatorClass cc = new CalculatorClass();
Integer product = cc.multiply(10, 5);
```

4. **List – Iteration**

   a. We want to do some calculations by passing a List of *decimal* as parameters.
   Create four different methods in the **OperationsHelper** class to do the following:

      i. Sum
      ii. Average
      iii. Highest
      iv. Lowest

      Every method must have an input parameter of type **List of decimal**.
      If the provided list is empty, then show a message like “Please provide values to do the operation” and return null otherwise return the calculated value.

5. **List Method: (optional)**

   Create a class named as ListPractice and fulfill the following requirements:

   a. Create a static class variable of type List and name it as planetsList here Planets is the class that we already have in our org.
   b. Create a method **addPlanet**.
   Take planetName, planetNumber and numberOfMoons as parameters in this method.

   Inside the method - Method should create the instance of the Planets class only when the planetName is provided and once the instance is created add it to the planetsList variable.
   This method should return the following messages:
   If planetName is provided and an instance is added to the planetsList variable, then return "**Planet added successfully**" message. if planetName is not provided then return "**Planet without a name cannot be added to the list**"

Create multiple instances of the ListPractice class, add planets with the help of addPlanet method and print the value in the planetsList variable in anonymous script.

6. **Maps – Iteration**
    1. Store the currency exchange rate values in a Map named ‘currISOCodeAndExchangeRateMap’ for any 5 currency ISO codes. Like below:
        a. AUD => 0.73
        b. CAD => 0.78
        c. EUR => 1.13
        d. NZD => 0.68
        e. INR => 0.013
    2. Fetch the currency conversion rate for any one currency ISO code and print it.
    3. Check if the above Map has a conversion rate for SAD ISO code.
    4. Collect all the currency ISO codes from currISOCodeAndExchangeRateMap for which the conversion rate is >= 1 in one variable and other iso codes in other variables.

        Expected Output: EUR – In one variable. AUD, CAD, NZD, and INR – In another variable.

END