Plan for the day:

1. MCQ for CSS
2. Subjective test for Core Java M1
3. Java hands on practice for M1
4. Cheat sheet for Java
5. SonarQube

Elasticity? How is it different from Scalability?

Elastic Scalability

Scalability is ability to handle growing or reducing needs using same infrastructure.

Elasticity is the ability to scale by provisioning more infra

Auto scaling

if(true)

System.out.println();

else

System.out.println();

Code quality:

1. {} for every true or false part of if and loop
2. Every method to have only 1 return statement
3. Every variable declared must be initialized
4. Any unused variables must be removed
5. Any code commented out needs to be removed
6. Package name is must for any class
7. Unused imports to be removed

SonarQube demo: demo is given and recorded the same and uploaded.

Demo is recorded as video:

<https://www.veed.io/view/f6be3162-57a5-48f5-94a9-e0b05f29fc67?panel=share>

and the link is added to the LMS.

Java practice challenge questions are uploaded to git.

Preparation for MCQ content is added to git.

MCQ on CSS test at 11:30 am

Subjective test on core java after lunch

**ArrayList to String Array**

Write a program that performs the following actions:  
  
1.Read m strings as input (fruit names).  
2.Create an arraylist to store the above m strings in this arraylist.  
3.Read n strings as input (fruit names).  
4.Create an arraylist to store the above n strings in this arraylist.  
5.Write a function fruitSelector which accepts the arraylists as input.  
6.Remove all fruits whose name ends with 'a' or 'e' from first arrayList and remove all fruits whose name begins  with 'm' or 'a' from second arrayList then combine the two lists and return the final output as a String array.  
7.If the array is empty the program will print as “No fruit found”  
Include a class UserMainCode with the static method **fruitSelector** which accepts the two arraylists and returns an array.  
  
Create a Class Main which would be used to read n strings and call the static method present in UserMainCode.  
  
**Input and Output Format:**  
  
Input consists of an integer (m) denoting the size of first arraylist. The next m elements would be the values of the first arraylist. The next input would be n denoting the size of the second arraylist. The next n elements would be the values of the second arraylist.  
  
Output consists of an array as per step 6. Refer sample output for formatting specifications.  
  
**Sample Input 1:**  
3  
Apple  
Cherry  
Grapes  
4  
Orange  
Mango  
Melon  
Apple  
**Sample Output 1:**  
Cherry  
Grapes  
Orange

HTML

Create

**Digits - II**

Write a program to read a non-negative integer n, compute the sum of its digits. If sum is greater than 9 repeat the process and calculate the sum once again until the final sum comes to single digit.Return the single digit.  
Include a class UserMainCode with a static method **getDigitSum** which accepts the integer value. The return type is integer.  
Create a Class Main which would be used to accept the string and call the static method present in UserMainCode.  
  
**Input and Output Format:**  
Input consists of a integer.  
Output consists of integer.  
Refer sample output for formatting specifications.  
  
**Sample Input 1:**  
9999  
**Sample Output 1:**  
9  
  
**Sample Input 2:**  
698  
**Sample Output 2:**  
5

**Add and Reverse**

Given an int array and a number as input, write a program to add all the elements in the array greater than the given number. Finally reverse the digits of the obtained sum and print it.

Include a class **UserMainCode** with a static method “**addAndReverse**” that accepts 2 arguments and returns an integer.The first argument corresponds to the integer array and the second argument corresponds to the number.

Create a class **Main** which would get the required input and call the static method **addAndReverse** present in the UserMainCode.

**Example:**

Input Array = {10,15,20,25,30,100}

Number = 15

sum = 20 + 25 + 30 + 100 = 175

output = 571

**Input and Output Format:**

The first line of the input consists of an integer that corresponds to the number of elements in the array.

The next n lines of the input consists of integers that correspond to the elements in the array.

The last line of the input consists of an integer that corresponds to the number.

Output consists of a single integer.

**Sample Input**

6

10

15

20

25

30

100

15

**Sample Output**

571

**Sum of Digits in a String**

Write code to get the sum of all the digits present in the given string.

Include a class **UserMainCode** with a static method **sumOfDigits** which accepts string input.

Return the sum as output. If there is no digit in the given string return -1 as output.

Create a class **Main** which would get the input and call the static method **sumOfDigits** present in the UserMainCode.

**Input and Output Format:**

Input consists of a string.

Output is a single integer which is the sum of digits in a given string.

Refer sample output for formatting specifications.

**Sample Input 1:**

good23bad4

**Sample Output 1:**

9

**Sample Input 2:**

good

**Sample Output 2:**

-1

**String processing – Long + Short + Long**

Obtain two strings S1,S2 from user as input. Your program should form a string of  “long+short+long”, with the shorter string inside of the longer String.  
Include a class UserMainCode with a static method **getCombo** which accepts two string variables. The return type is the string.  
Create a Class Main which would be used to accept two Input strings and call the static method present in UserMainCode.  
**Input and Output Format:**  
Input consists of two strings with maximum size of 100 characters.  
Output consists of an string.  
Refer sample output for formatting specifications.  
**Sample Input 1:**  
Hello  
Hi  
**Sample Output 1:**  
HelloHiHello

**Password Generator**

Write a Java program that accepts the first name, lastname, age as input and display their password.

The password must be combination of first 3 letters of firstname, last 3 letters of lastname and @ symbol then their age. The first letter should be capitalized.

Example  
First Name: raja

Last Name: siva

Age: 22

Output:

Rajiva@22

<https://docs.google.com/spreadsheets/d/1QZoA3iLkcvezVNC4xDFtKp4gX3j1ATfEuPnRsEjuKFA/edit?usp=sharing>