CSS

HTML5

Subjective test html+css, core java, SQL(1 Qn from SELECT)

Bootstrap

CSS

We have learnt

Syntax

Selectors

How to inline, internal, external

Box model

Task:

CSS tables

Getting started with Bootstrap:

What is bootstrap?

Is a CDN and open source, many providers like Google, W3Schools etc are providing their version of Bootstrap which is css, jquery that can be applied to our html page to enhance the appearance and effects of the UI.

Bootstrap typography

When the bootstrap CDN is applied to a html file, a default style is applied to the elements.

Primary

Success

Info

Warning

Danger

<https://www.w3schools.com/bootstrap4/tryit.asp?filename=trybs_txt_bgcolors&stacked=h>

Grid Model

Module 1 Revision:

UML

Unified Modelling Language

UML diagrams are used to represent the entire software artificats that are going to be constructed.

Before start coding, these diagrams are drawn to keep a record of what we exactly promised to do.

UML is an unified version of Rambaugh, Jacobson and Booch modelling techniques.

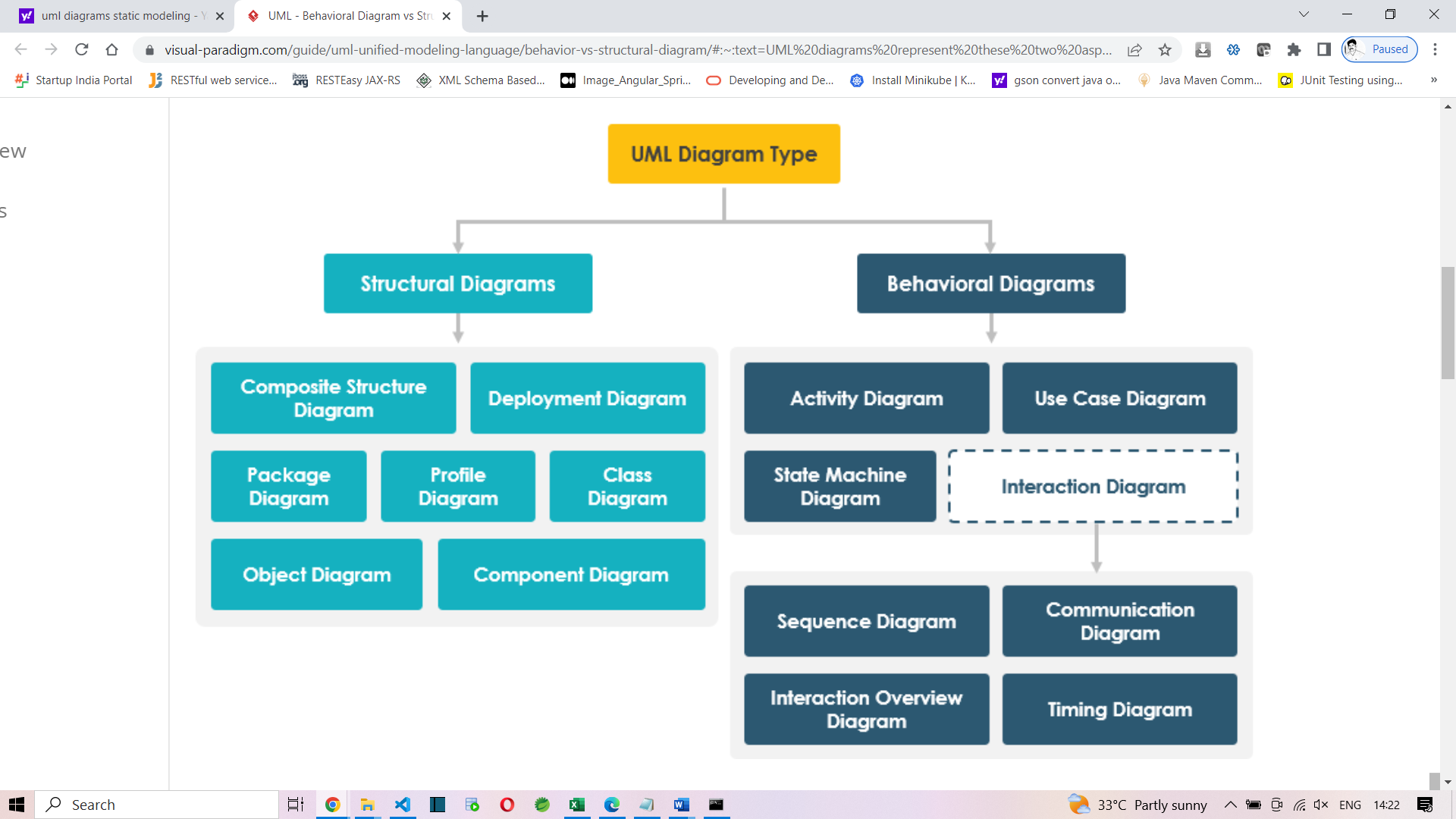
UML diagrams are categorized into:

Static / Structural

Dynamic / Behavioral

In UML 2.2 there are 14 types of UML diagrams, which are divided into these two categories:

* 7 diagram types represent **structural** information
* Another 7 represents general UML diagram types for **behavioral** modeling, including four that represent different aspects of interactions.



Callenge:

**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

Solution:

**import** java.util.Scanner;

**public** **class** App {

**public** **static** **int** digitSum(**int** num)

{

**int** sum=0;

**if**(num>9)

{

//get the digits and sum

String str=num+"";

sum=0;

**for**(**char** c : str.toCharArray())

{

**int** digit=Character.*getNumericValue*(c);

sum+=digit;

}

**if**(sum>9)

**return** *digitSum*(sum);

}

**return** sum;

}

**public** **static** **void** main(String[] args) {

Scanner sc=**new** Scanner(System.***in***);

**int** input=sc.nextInt();

System.***out***.println(*digitSum*(input));

}

}

**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

Solution:

**import** java.util.Scanner;

**public** **class** App {

**public** **static** **void** main(String[] args) {

Scanner sc=**new** Scanner(System.***in***);

**int** n=sc.nextInt();

**int** arr[]=**new** **int**[n];

**for**(**int** i=0;i<n;i++)

{

arr[i]=sc.nextInt();

}

**int** num=sc.nextInt();

**int** sum=0;

**for**(**int** i=0;i<n;i++)

{

**if**(arr[i]>num)

sum+=arr[i];

}

//reverse the sum

StringBuilder sb=**new** StringBuilder(sum+"");

sb.reverse();

**int** result=Integer.*parseInt*(sb.toString());

System.***out***.println(result);

}

}

**Flush Characters**

Write a program to read a string from the user and remove all the alphabets and spaces from the String, and**only store special characters and digit** in the output String. Print the output string.

Include a class **UserMainCode** with a static method **getSpecialChar**which accepts a string. The return type (String) should return the character removed string.

Create a Class Main which would be used to accept a string and call the static method present in UserMainCode.

**Input and Output Format:**

Input consists of a strings.

Output consists of an String (character removed string).

Refer sample output for formatting specifications.

**Sample Input :**

cogniz$#45Ant

**Sample Output :**

$#45

Solution:

import java.io.\*;

import java.text.SimpleDateFormat;

import java.util.\*;

public class Main {

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

String s1=sc.next();

String s2=s1.replaceAll("[a-zA-Z]", "");

System.out.println(s2);

}

}

---------------Long route------

**import** java.util.regex.Matcher;

**import** java.util.regex.Pattern;

**public** **class** App2 {

**public** **static** **void** main(String[] args) {

String input="P@$$w0rd";

StringBuilder sb=**new** StringBuilder(input);

Pattern p = Pattern.*compile*("[0-9A-Za-z]+");

Matcher m = p.matcher(input);

**while**(m.find())

{

System.***out***.println(m.group());

**int** index=sb.indexOf(m.group());

**int** length=m.group().length();

System.***out***.println(index+" to "+(index+length));

sb.delete(index, (index+length));

}

System.***out***.println("Result="+sb.toString());

}

}

Another route:

Character.isLetter

Character.isDigit

Other than that is symbol

Discussion on Module 1 topics:

Entity classes, Serialization/De-Serialization

●Error/ Exceptions Handling, Custom Exceptions

●Reading and writing data from files, JSON, and XML

●

CUSTOMER PURCHASE PRODUCT

CUSTOMER

PRODUCT

in db:

CUSTOMER

PRODUCT

these are tables

in java:

Customer

Product

Object - Relational mapping

ORM

select \* from HR.EMPLOYEES;

100 Steven King 24000

101 Neena Kochar 17000

List<Employee>

Serialization

is a process of converting an object into a series of bytes.

Deserialization

is vice versa

java -ea App

java -da App

interoperable