1. Adding all the array values at a time – **Arrays.asList(10,20,30);**

-> ArrayList<Integer> number = new ArrayList<Integer> (Arrays.asList (10, 20, 30));

1. we can directly print ArrayList –

-> SYSO (number);

1. What is literals – when we initialize static array without ‘new’ Keyword.

-> int a[] = {1,2,3,4};

1. When you print static array directly, it will print memory address.

-> 1. int a[] = {1,2,3,4}; //literals

OR

-> 2. int a[] = new int[2];

a[0]=1;

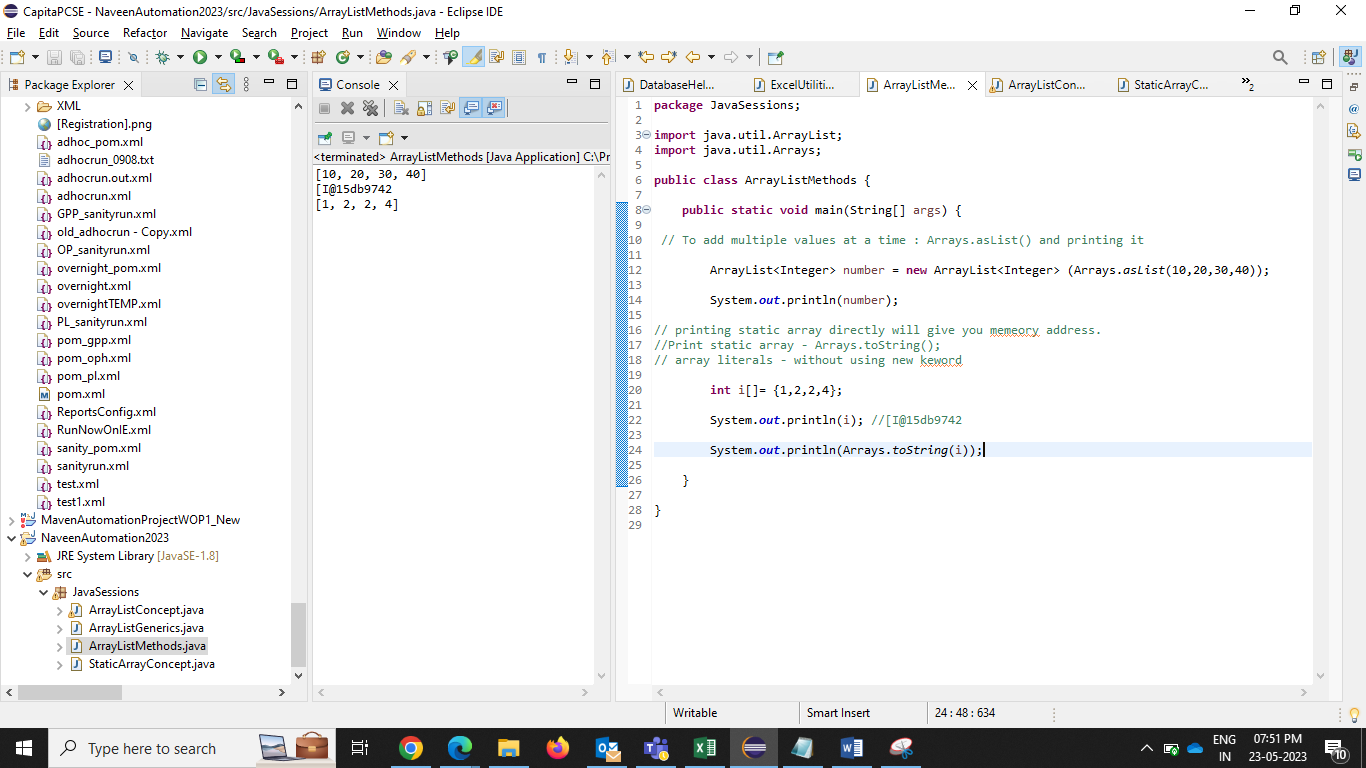
a[1]=1;

-> SYSO(a); // in above both ex. Ouput will be - [I@15db9742

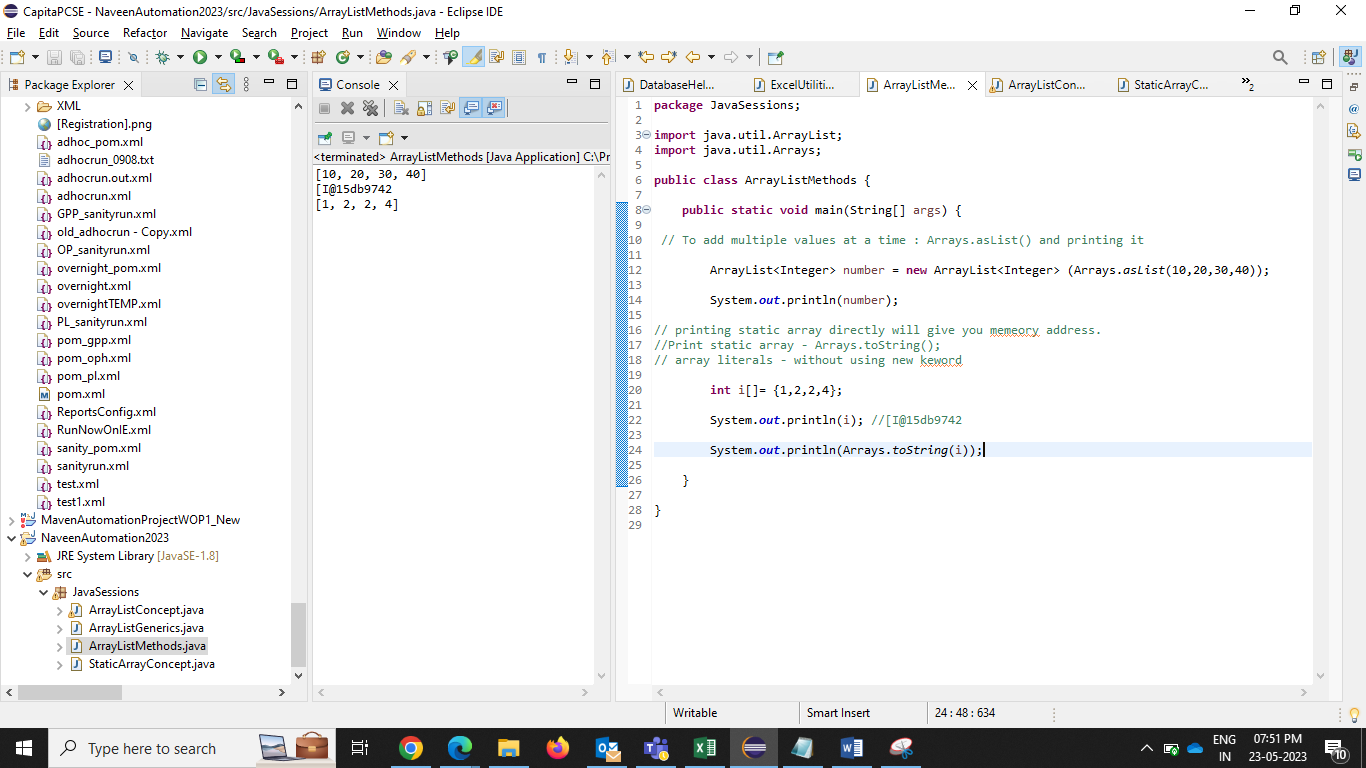
So, to print static Array we use **for loop** or **mentioned index a[i]** while printing.

Also, to overcome problem no. 4 , we use method = **Arrays.toString(i);**

**Incase of dynamic array, we can directly print it.**



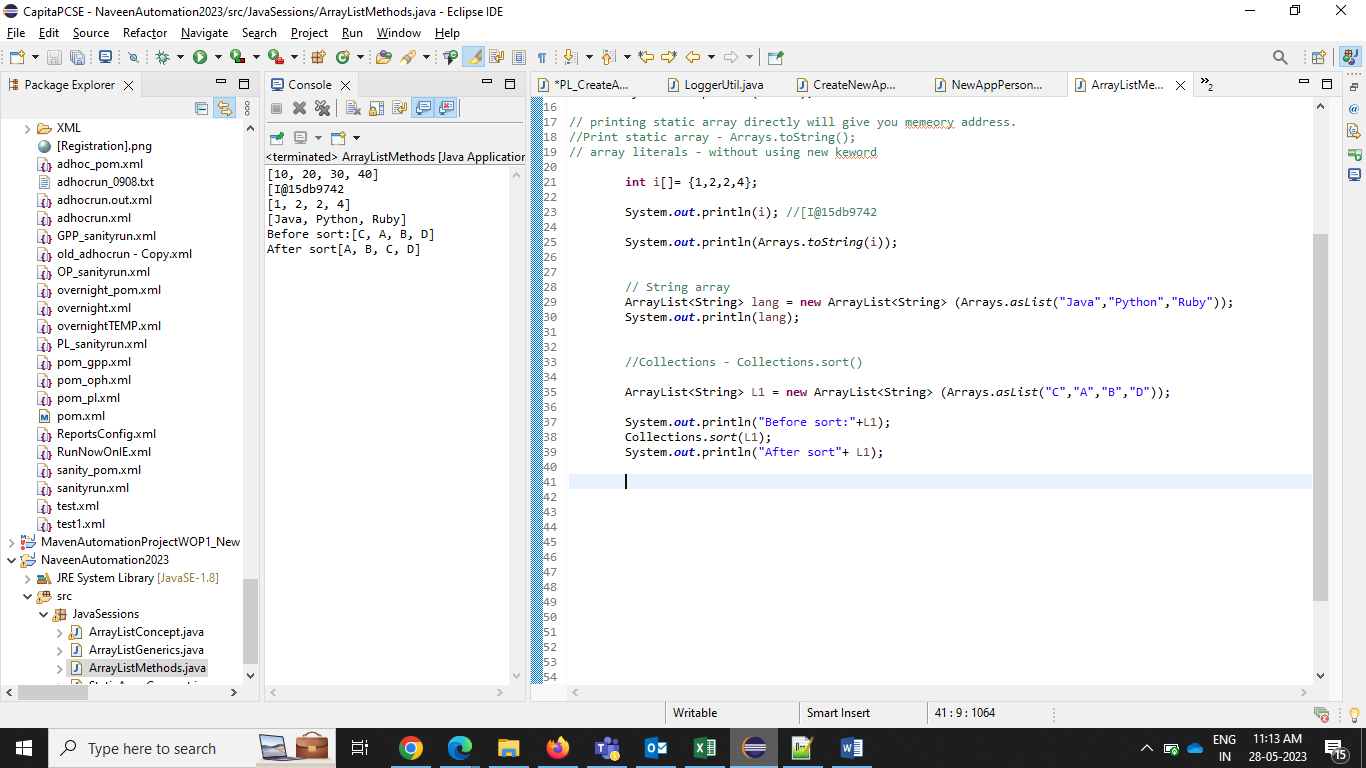
String ArrayList –



**Collections class - to sort ArrayList - Collections.sort();**

ArrayList<String> L1 = new ArrayList<String> (Arrays.asList(“A”,”B”,”C”));

Collenctions.sort(L1);



**Compare 2 Array List - .equals() - will return Boolean value**

**L1.equals(L2) –**

ArrayList<String> L4 = **new** ArrayList<String>(Arrays.*asList*("P","Q","R","S","Z"));

ArrayList<String> L5 = **new** ArrayList<String>(Arrays.*asList*("P","Q","R","S","T"));

/\*all the common elements from L4 will be removed, comparing with L5 and uncommon element will printed, Z \*/

L4.removeAll(L5);

System.***out***.println(L4); // Z

ArrayList<String> L44 = **new** ArrayList<String>(Arrays.*asList*("P","Q","R","S","Z"));

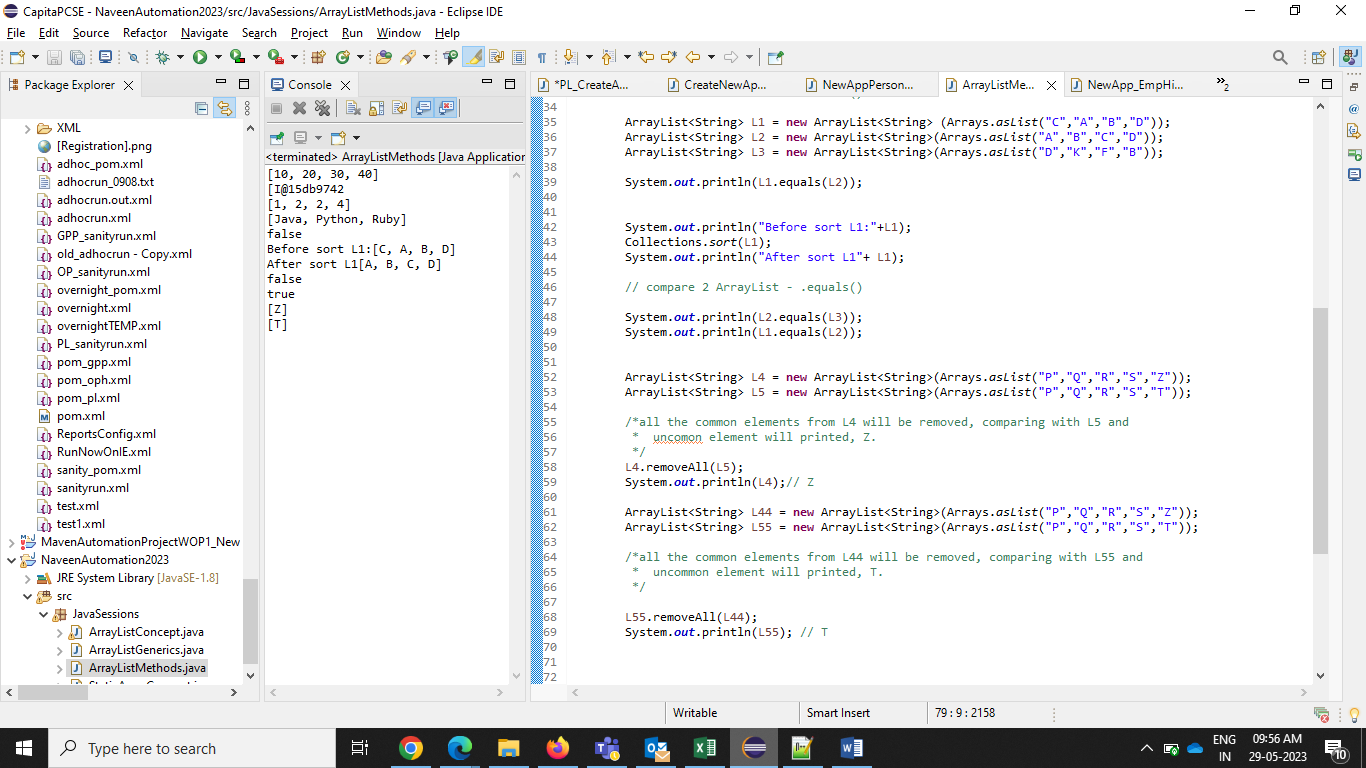
ArrayList<String> L55 = **new** ArrayList<String>(Arrays.*asList*("P","Q","R","S","T"));

/\*all the common elements from L44 will be removed, comparing with L55 and

uncommon element will printed, T \*/

L55.removeAll(L44);

System.***out***.println(L55); // T



**//** To find out **common** element - L1.retainAll(L2)

**// To find out common element**

/\*all the uncommon elements from J1 will be remove, comparing with J2 and

\* common element will printed from J1, [JAVA, PYTHON, PHP, C++] \*/

ArrayList<String>=

**new**ArrayList<String>(Arrays.*asList*("JAVA","PYTHON","PHP","C++",".Net"));

ArrayList<String> J2 = **new** ArrayList<String>(Arrays.*asList*("JAVA","PYTHON","PHP","C++","Ruby"));

J1.retainAll(J2);

System.***out***.println(J1);//[JAVA, PYTHON, PHP, C++]

/\*all the uncommon elements from J22 will be removed, comparing with J11 and

\* common element will printed from J22, [JAVA, PYTHON, PHP, C++]

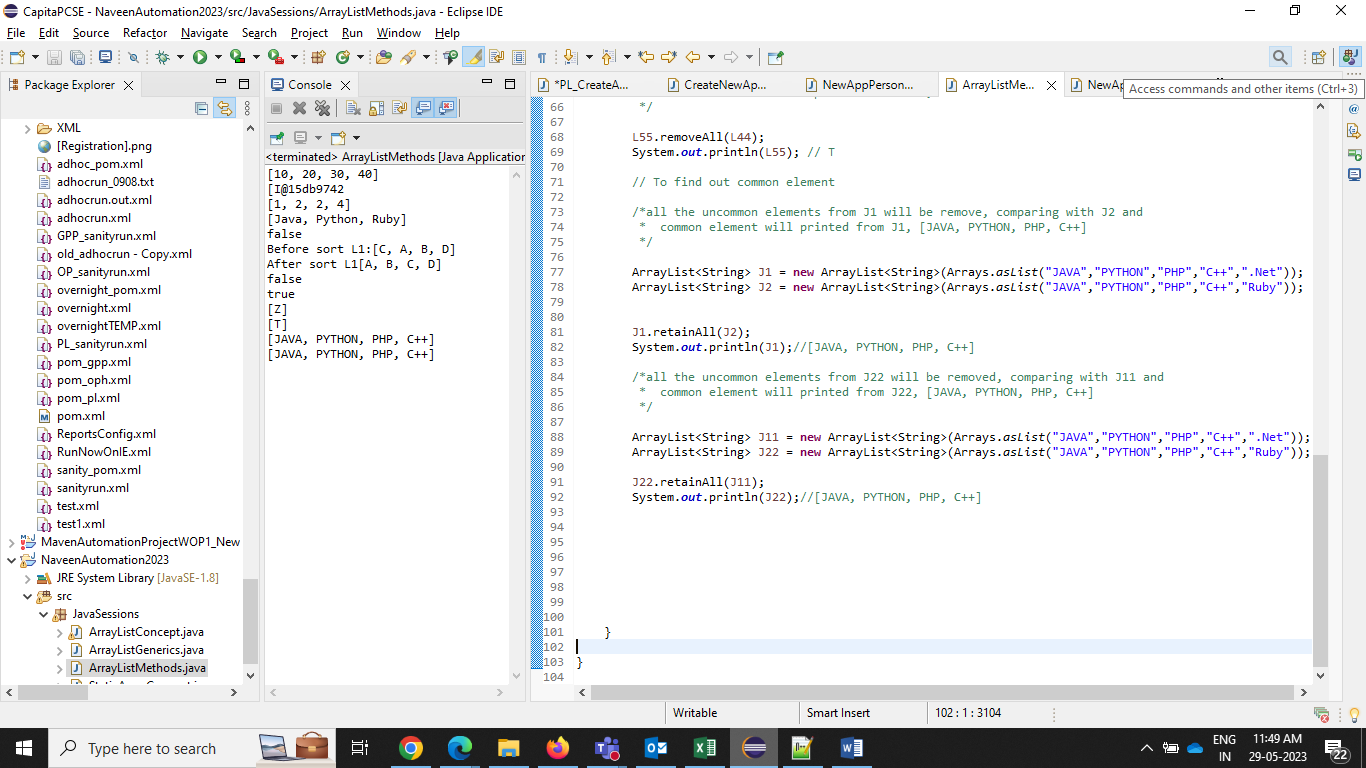
\*/

ArrayList<String> J11 = **new** ArrayList<String>(Arrays.*asList*("JAVA","PYTHON","PHP","C++",".Net"));

ArrayList<String> J22 = **new** ArrayList<String>(Arrays.*asList*("JAVA","PYTHON","PHP","C++","Ruby"));

J22.retainAll(J11);

System.***out***.println(J22);//[JAVA, PYTHON, PHP, C++]



// adding element from 1 array to another - .addAll()

ArrayList<String> N1 = **new** ArrayList<String>(Arrays.*asList*("Sneha","Anvi"));

ArrayList<String> N2 = **new** ArrayList<String>(Arrays.*asList*("Priya","Rekha"));

N1.addAll(N2);

System.***out***.println(N1); // all the values from N2 will be added to N1

System.***out***.println(N2);

//making copy of ArrayList - .clone()

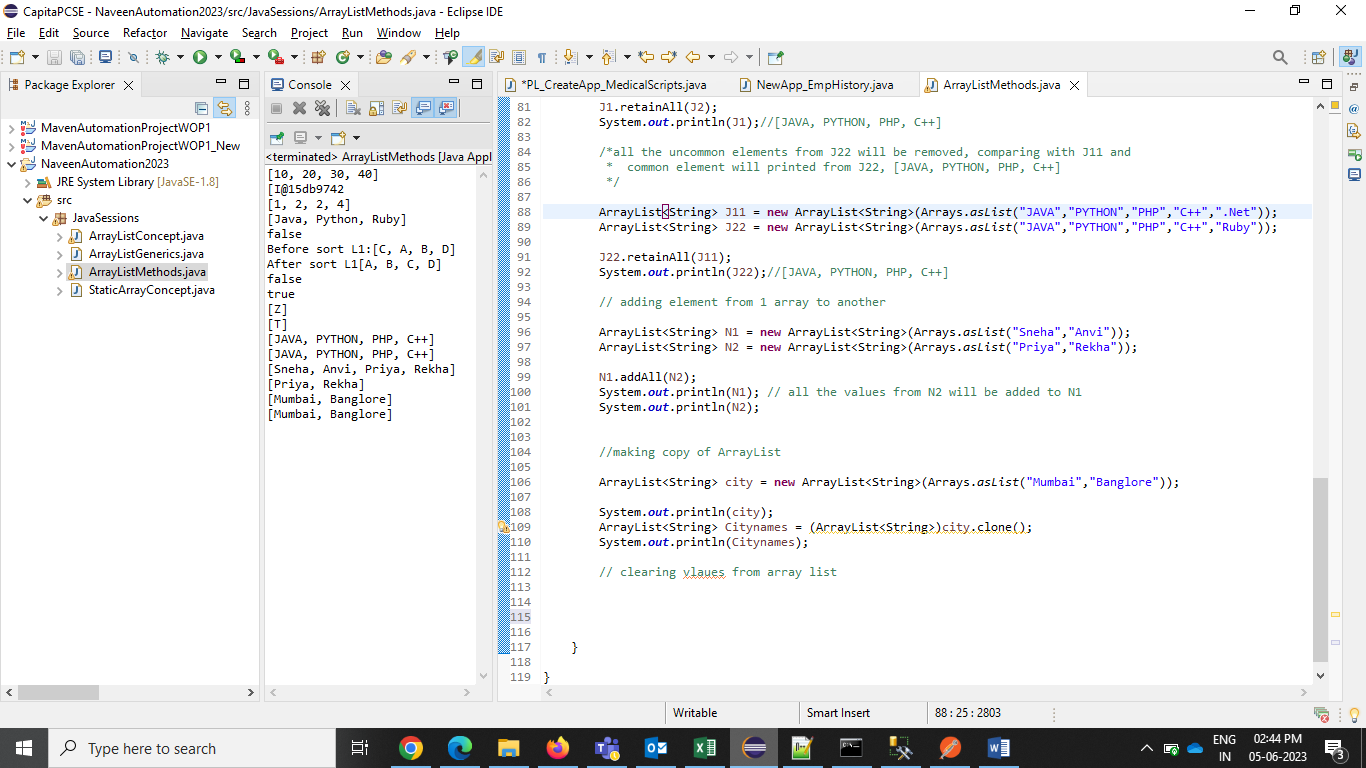
ArrayList<String> city = **new** ArrayList<String>(Arrays.*asList*("Mumbai","Banglore"));

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

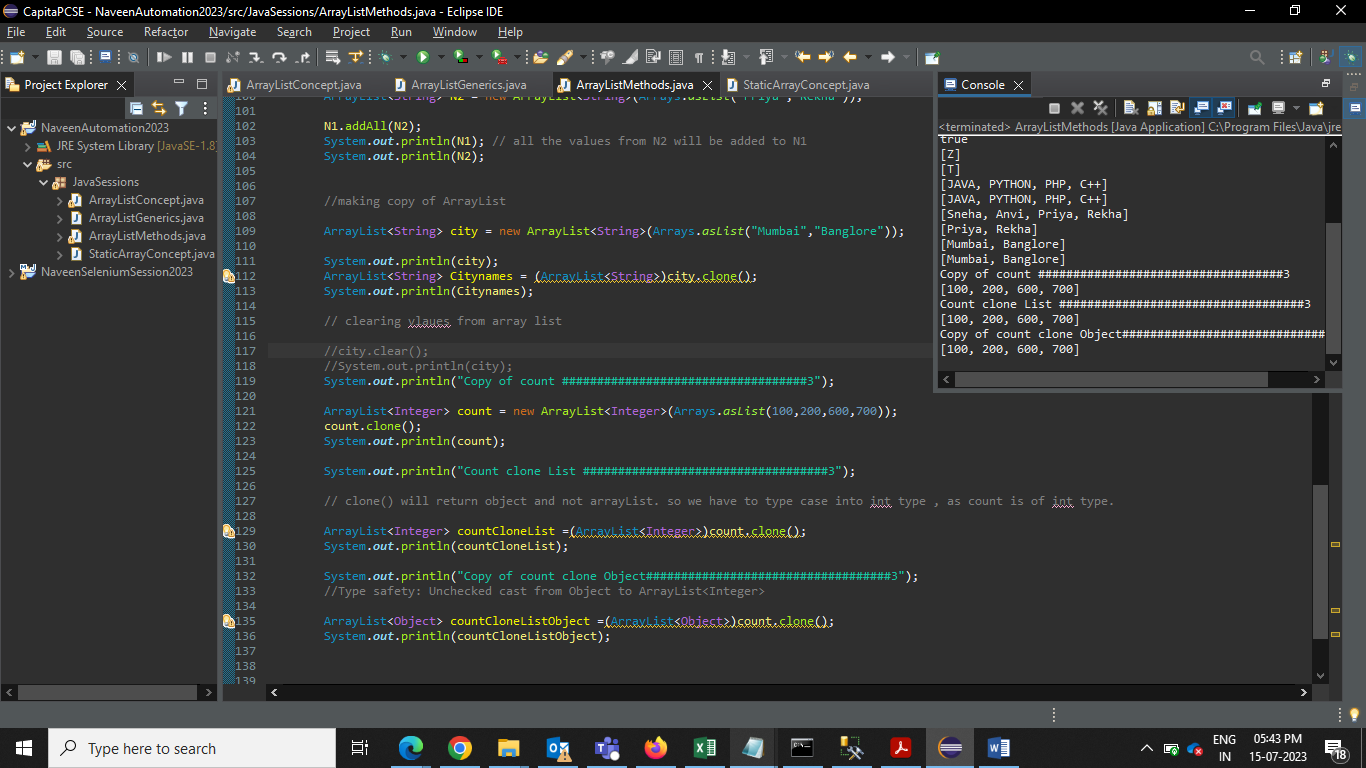
ArrayList<String> Citynames = (ArrayList<String>)city.clone();

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

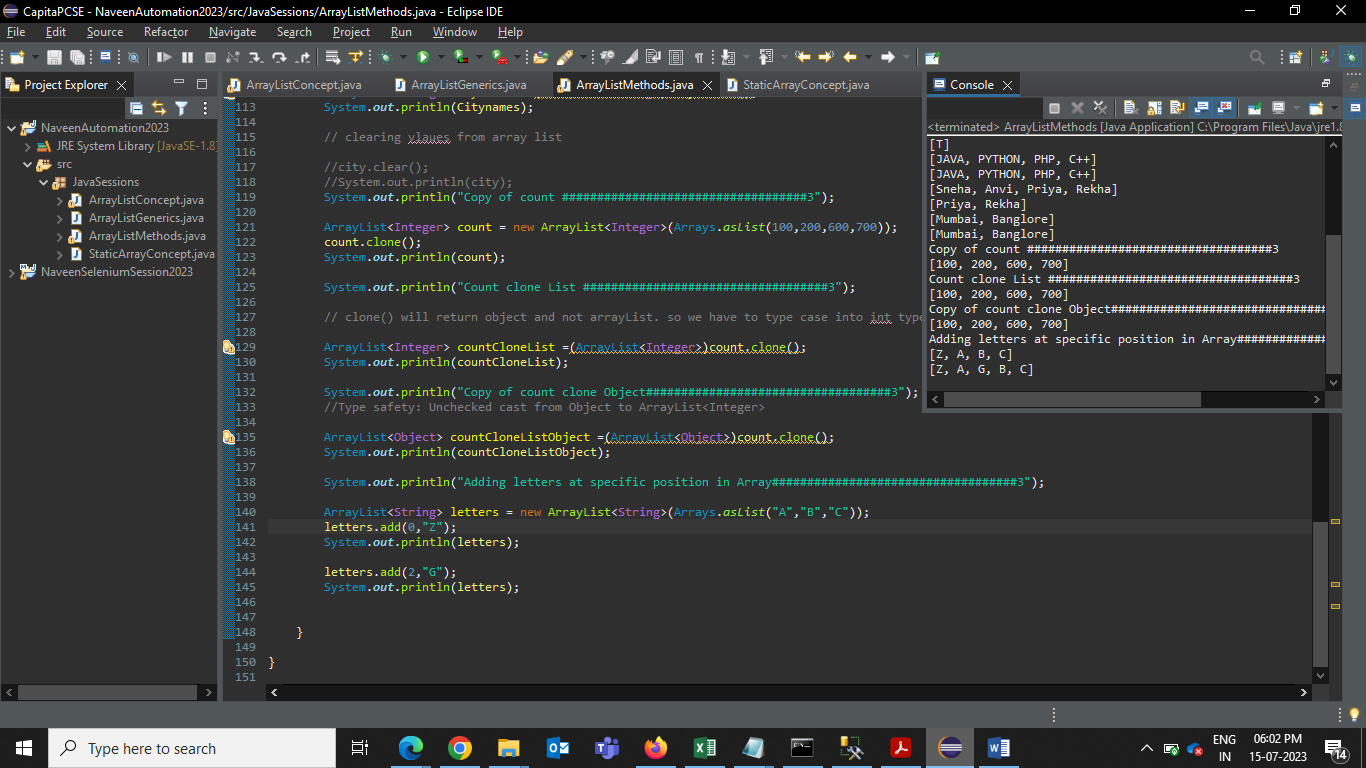
//To clear values form ArrayList - .clear()



.clone() – this method will return object and not a ArrayList.. so we have to type cast it.



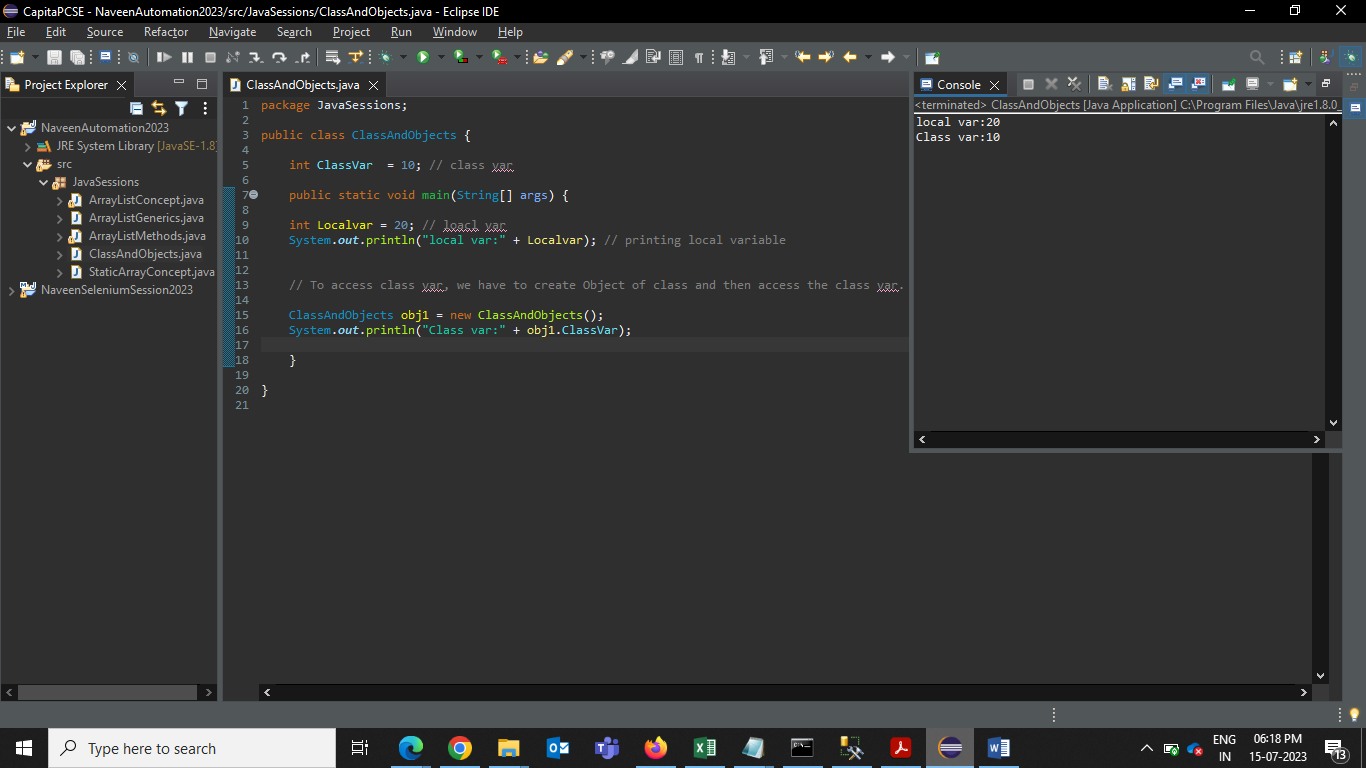
Adding values In arrayslist at specific index –



Class variable and Local variable –

Class var – scope is within class. To access class var, create obj of the class

Local var – scope is within method. We can access directly local var



//Class - template / blueprint / category for the objects

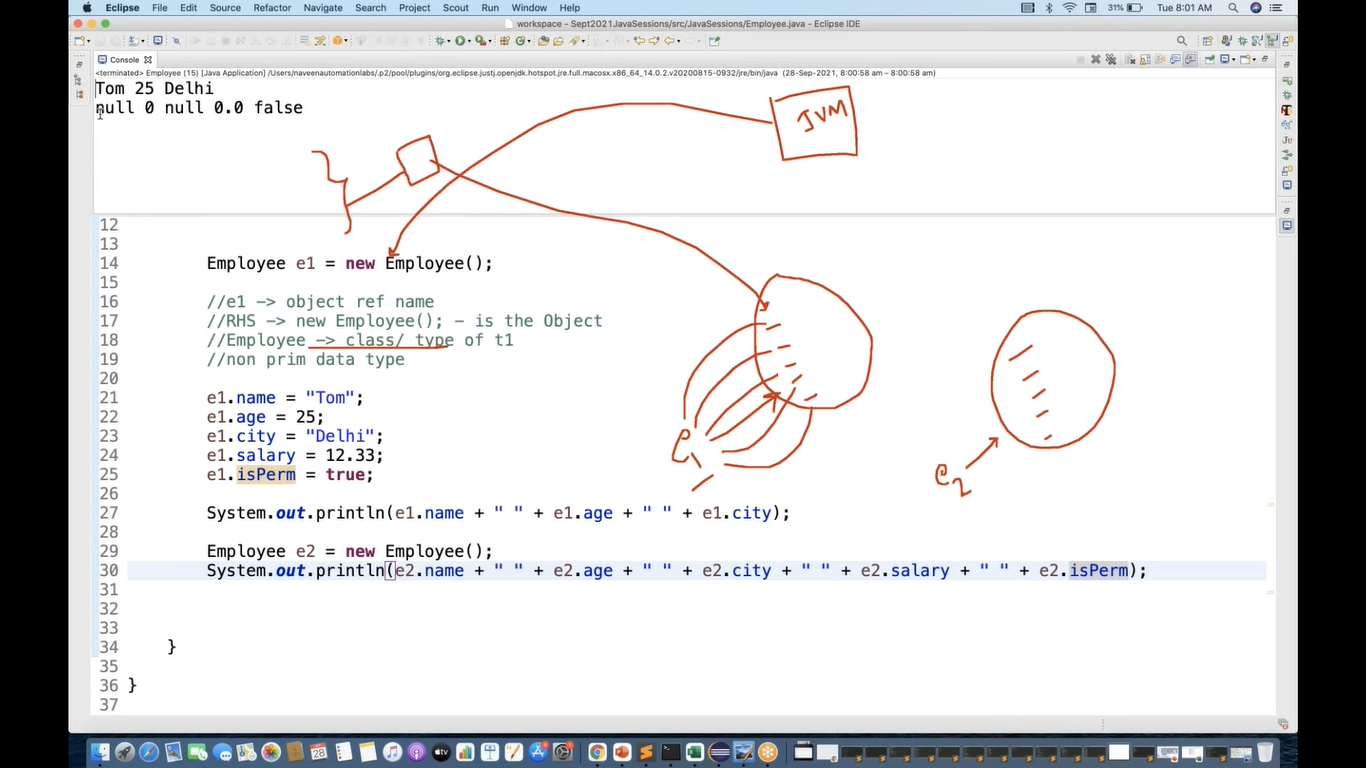
//example- human, Animal, employee

//Object - objects are always physical entity and should be created from class template/blueprint

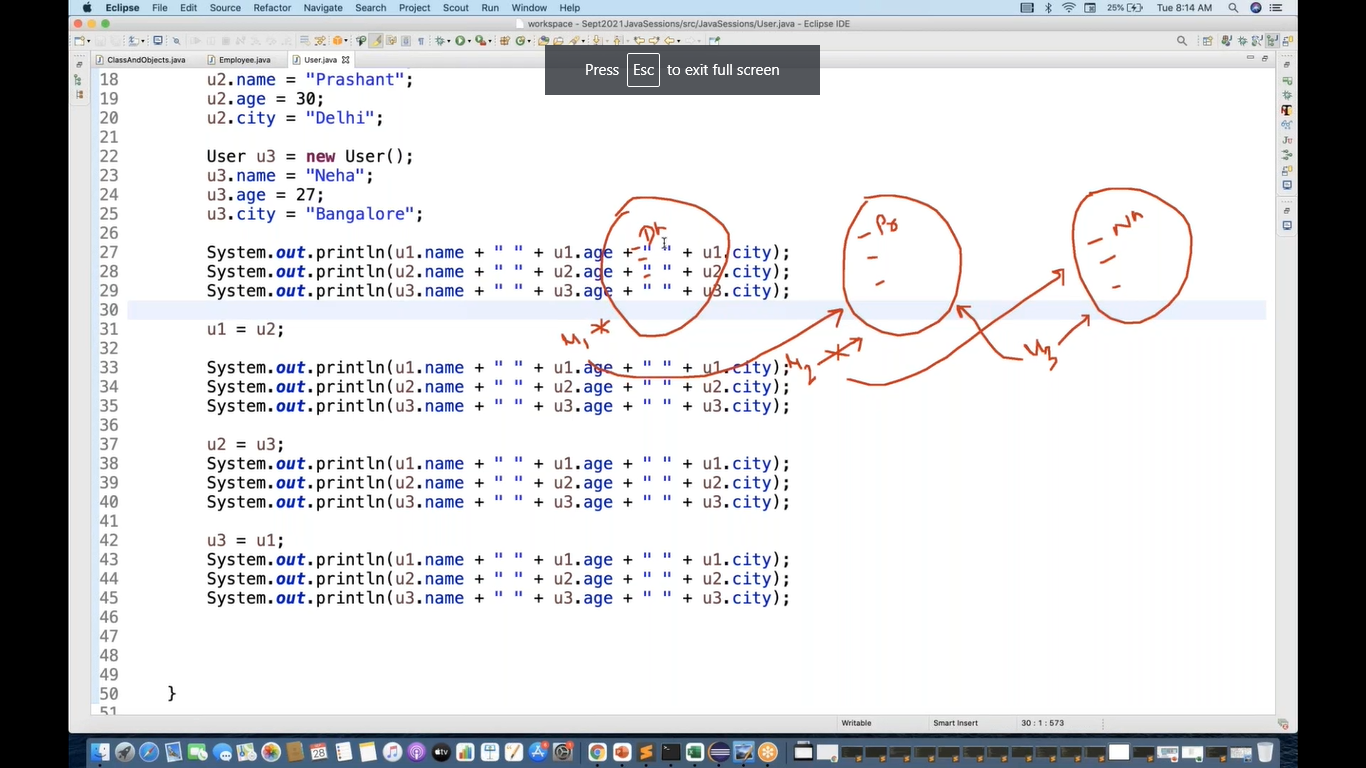
// example- human body, diff animals, diff employee with diff emp info.

// Class will have certain properties which will help us to create the objects.

Default values will be printed when we do not define variable-



Object reference concept –



Output –

