Consolidated Questions:

1. **Coding exercise**- Set of first names and last names and we have to generate the person names randomly and display the duplicate person name who has a similar first name and last name
    a. **Follow up question** – What will be the time complexity?
2. **Coding exercise**: Given one Array with some values (3, 1, 2, 3, 4, 5) and need to remove the value randomly of user's choice
3. WAP to take an input of List of Strings and returns a list of unique strings
4. WAP to remove duplicates from an array
5. WAP to sort an array using Collections
6. WAP to sort an array without using collections or using array sort ()
7. Find out the missing number in the array of length n-1 and integers in the range of 1 to n.
8. Move the zero elements to the end of array
    a. input --> int a[]={0,1,2,0,3,0,0};
    b. output--> int a[]={1,2,3,0,0,0,0};
9. Sort the given array [2,1,1,0,0,0,3]
10. Count the no.of occurences each digit in given array [1,1,1,3,2,2,2,2,2,0,0,0]
11. WAP to remove duplicates from the given array without using set
12. Given char array [a,b,b,c,d,f]
    a. given value k
    b. return first element that occurs in this array k times
    c. k=2 return b
    d. k=3 return anything
    e. k=1 return a

13. Find the duplicate character in the given String "abcdedef"

14. Search particular character in the given string

    a. String = "Manjusha"

    b. Our Target Value = "u" then returns Test Passed

    c. Our Target Value = "K" then returns Test Failed

15. find the first repetitive letter from the given string String input="abcddeff". return 'd'
16. Given array [0,1,2,3,5,6,7]
    a. if target value=5 return 3
    b. if target value=4 return 3;

17. input="(())" return true input="((())" return false. if String contains matched Open and close brackets return true otherwise return false.
18. Given an array (1,3,4,5,2) and a number k, find all elements that appear k times
19. Given an array of (1,2,3,4,5,2) and find the no of occurrences in an unsorted array
20. Given a sorted array and a target value, return the index if the target is found. If not, return the index where it would be if it were inserted in order.
    a. [1,3,6,8,9] target= 3 return the index of 3
    b. target = 7 return the index if target will be inserted in the array (sorted)
21. Let's say we have an array
    a. [2,5,6,7,8] target=6 -> return index of the element 6
    b. [3,5,6,9] target=4 -> return index of element 4
22. Find out the missing number in the array of length n-1 and integers in the range of 1 to n.
23. Given a sorted integer array and a target value, return the index if the target is found. If not, return the index where it would be if it were inserted in order.
    a. [1,3,5,6,7] k=4 -> return the index for 4
    b. [2,4,5,7,8] k=5 ->5's index

24. WAP to Print the reverse of a string
    a. Input = “Chennai is a green city”
25. WAP to show the implementation of Hash Map
26. loop through the array and print all the elements in the reverse order
27. Print the word count in a given sentence
28. String s = "This is sample string to count the strings"
29. Write a program to find the duplicate characters in a string?
30. Write a program to count:
    a. No. of characters in a string
    b. No. of repeated words in a sentence

31. Write a program to add the integers in a string

    a. String s = "abc123def45gh6i7"

    b. Output: 123+45+6+7 = 181

32. String colors = "red, blue, green, red, blue, orange, blue, red"
    a. Write a program to find the words number of time repeating using HashMap.
33. String s ="web internet web chrome internet safari"
34. String "racecar", "abcdecac", "override", "madam"
    a. Find the count of each character in the above metioned strings
35. Find the palindrome terms and if its a palindrome get the count of first character in the string and if its not a palindrome find the highest count of a character in the string

36. Find the first non-repeating character in a string by iterating only once through the string with best space and time complexity
37. For the given string count the reported words using hash map
    a. “Book key table key smart book key on the table science book”
38. Make given array as unique and display above (2.5<N) only.

    =====Example=======

    Given array is {1,2,3,8,5,4,2,5,9,7}

    Op Would be >> {3,8,5,4,9,7}.
39. method should retun boolean value By taking string values and as shown below.

    S=“rabbbit”;

    T=“rabbit”//True

    U=“tabbit”//false

    V=“ab”//true

40. Write a program to add the integers in a string

    String s = "abc123def45gh6i7"

    Output: 123+45+6+7 = 181

    Write Unit tests to test the given program

    What is the Complexity of the program written and how it can be improved?

41. Given two Strings, find if the second string is a sub-string of the first. (Both in a single line as well as without inbuilt functions)

42. Given a string find the total number of sub-strings that can be formed out of it. (This is a mathematical problem of finding the number of permutations of a given number.) also write a program to print all the sub-strings.
43. Given a range of numbers e.g. (1,4,5,8,6,2,7,3,9). write a program to find pairs of numbers whose sum is 8. Also what approach can be applied if the same range is scaled to over 10 million numbers (The trick here is to distribute the range into smaller range and send it for parallel execution and merge the outputs)

Follow up questions:

1. Differences between equals method and == operator
2. Example to demonstrate override and overload
3. How & where will I use interfaces and classes
4. Rest apis, different methods etc.
5. What are access modifiers
6. What is the difference between public, private, protected and default?
7. Why do we need static access modifier?

Problem solving:

Below codes will compile ?

1.
```
public class Test{
void m1(){
System.out.println("From m1");
}void m2 ()
{System.out.println("From m2");
}static void m2 ()
{ System.out.println("inside static");
   m1 ();
}private void m3()
{System.out.println("From m3");
}}
```

How to override / overload from above code?

```
1. String s1 = “java”;
s1 = s1.concat(“ knowledge”);
System.out.println (“Value of s1 is “ + s1);
```

```
2. public static void main(String[] args){
 String s1 = "abc";
 String s2 = "abc";
 System.out.println(s1 == s2);
 String s3 = "abc";
 String s4 = new String("abc");
 s3 = s3.intern();
 s4 = s4.intern();
 System.out.println(s3.equals(s4)); }
```

```
3. class BaseClass {
public String method() {
return "I am from base class"; }}
class Childclass extends BaseClass{
private static int counter = 0;
public String method(int x) {
return "Derived Class - Childclass";}
public static void main(String[] args) {
ChildClasseClass bg = new BaseClass();
System.out.println(bg.method());} }
```

```
     4.  class A
{String s = "Class A";}
class B extends A
{String s = "Class B"; {
 System.out.println(super.s);}}
class C extends B{
String s = "Class C";{
 System.out.println(super.s); }
}public class MainClass
{ public static void main(String[] args){
  C c = new C();
 System.out.println(c.s);
   }
}
```

```
     5.  public class B{
B b= new B();
public int show(){
return (true ? null : 0);
}
public static void main(String[] args)  {
B b= new B();
b.show();
}
}
```

```
     6.  class A
```

```
{
static int method1(int i)
{
return method2(i *= 11);
}
static int method2(int i)
{
return method3(i /= 11);
}
static int method3(int i)
{
return method4(i -= 11);
}

static int method4(int i)
{
return i += 11;
}
public static void main(String [] args)
{
System.out.println(method1(11));
}
}
-------------------
```

7. class X

```
{
static int x = 1234;
static class Y
{
static int y = x++;
static class Z
{
static int z = y++;
}
}
}
public class MainClass
{
public static void main(String[] args)
{
System.out.println(X.x);//1234

System.out.println(X.Y.y);//1234

System.out.println(X.Y.Z.z);//1234
}
}
```

----

8. public class swap_number

```
{
public static void main(String args[]){
int x = 10;
int y= 20 ;
x = x+y; //30
y = x-y;// 10
x =x-y; //20
}}
```

----

9. public class Test{

```
public void m1(){
}
private void m2(){
}
protected void m3(){
}void m4(){  }
static void m5(){
}}
class Test2 extends Test
{
Test2 t = new Test();
t.m1();}
```

```
11.  String s ="JAVA";
s.replace('A','C');
System.out.println(s);
```

```
12. public String reverseString(String s){
String s2 = new String();
for(int  i =s.length()-1; i> = 0 ; i--)
{
s2 = s1.subString(i, i-1);
}
return s2;
}
```

```
13. String s = “Nagpur ”;
s.concat(” M”);
System.out.println(s);
s = s.concat(” Nagpur M”);
System.out.println(s);
```

14.

```
String s1 = "abc";
String s2 = "abc";
System.out.println(s1.Intern()));
String s3 = new String("abc");
String s4 = new String("abc");
System.out.println(s3 == s4);
```

15.

```
public class B{
B b= new B();
public int show(){
return (8==8? 5: 0);
}
public static void main(String[] args)  {
B b= new B();
b.show();
}
}
```

16.

```
public class A{
public static void show(){
System.out.println("Static method called");
}
public static void main(String[] args)  {
A obj=null;
obj.show();}}
```

17.

```
int i = 9;
static {
sop(i);
}
```

18.

```
class X
{
static int x = 765;
static class Y
{
static int y = x++;
static class Z
{
static int z = y++;}}
}
public class MainClass
{
public static void main(String[] args)
{
System.out.println(X.x);

System.out.println(X.Y.y);

System.out.println(X.Y.Z.z);
}
}
```

Important Topics:

- OOPs, concepts with practical examples and implementation with respect to Automation Framework. Deep into Abstract Class and Interface.
- Thorough preparation on Access specifiers or modifiers(static and final with more focus). Their usages at different levels and significance.
- String Manipulations -String, StringBuilder and StringBuffer. When to choose what. Examples and algorithms on Strings.
- Collections: Classifications of all the implementations of List, Set, and Map. Should know when to use what(without exception). Example, ArrayList or LinkedList?? Should be capable of explaining all the pros and cons of each.
- Basic Algorithms or examples on Collections. Ex: In a given String calculate how many times a term occurred.
- Basics on Exception-handling. Multithreading and Generics.
- Better to know-> Hashing Mechanism, Custom Object Sorting(Comparable and Comparator), Object Cloning, Serialization etc..