**It is a part of the Java programming language that one can use for developing or creating a general-purpose app. Its main focus is to build such general applications. The J2SE (Java Standard Edition) is known as Core Java. It mainly covers concepts of object-oriented programming (OOP)**

These are some question

Q-01 : **How to reverse a String in java? Can you write a program without using any java inbuilt methods?**

ANS = There are many ways to do it, some of them are:

Using for loop

Using String Buffer

**Code**;-

**package org.arpit.java2blog;**

**public class ReverseStringForMain {**

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

**String blogName = "Virtusa training";**

**String reverse = "";**

**for (int i = blogName.length() - 1; i >= 0; i--) {**

**reverse = reverse + blogName.charAt(i);**

**}**

**System.out.println("Reverse of Virtusa training is: " + reverse);**

**}**

**}**

Graphical user interface, text, application, Word

Description automatically generated

**Q-02 : How to find duplicate characters in String in java?**

ANS = 1. Create a HashMap and character of String will be inserted as key and its count as value.

2.If HashMap already contains char, increase its count by 1, else put char in HashMap.

3.If value of Char is more than 1, that means it is duplicate character in that String.

**Code; -**

**public class DuplicateCharacters {**

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

**String string1 = "Great responsibility";**

**int count;**

**char string[] = string1.toCharArray();**

**System.out.println("Duplicate characters in a given string: ");**

**for(int i = 0; i <string.length; i++) {**

**count = 1;**

**for(int j = i+1; j <string.length; j++) {**

**if(string[i] == string[j] && string[i] != ' ') {**

**count++;**

**string[j] = '0';**

**}**

**}**

**if(count > 1 && string[i] != '0')**

**System.out.println(string[i]);**

**}**

**}**

**}**

**Text

Description automatically generated**

**Q-03: Separate odd and even numbers in an array?**

ANS = **1**. Initialise two index variable, left=0 and right=arr.length-1

**2**.Increment left variable until you get odd number

**3**.Decrement right variable until you get even number.

**3**.If left < right, swap arr[left] and arr[right]

**4**.In the end, you will see that you have even numbers on left side and odd numbers on right sides.

**Code; -**

**public class SeparateOddEvenMain {**

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

**int arr[]={12, 17, 70, 15, 22, 65, 21, 90};**

**System.out.println("Original Array: ");**

**for (int i = 0; i < arr.length; i++) {**

**System.out.print(arr[i]+" ");**

**}**

**arr=separateEvenOddNumbers(arr);**

**System.out.println("Array after separating even and odd numbers : ");**

**for (int i = 0; i < arr.length; i++) {**

**System.out.print(arr[i]+" ");**

**}**

**}**

**public static int[] separateEvenOddNumbers(int arr[])**

**{**

**int left=0;**

**int right=arr.length-1;**

**for (int i = 0; i < arr.length; i++) {**

**while(arr[left]%2==0)**

**{**

**left++;**

**}**

**while(arr[right]%2==1)**

**{**

**right--;**

**}**

**if(left<right)**

**{**

**int temp=arr[left];**

**arr[left]=arr[right];**

**arr[right]=temp;**

**}**

**}**

**return arr;**

**}**

**}Text

Description automatically generated**

**Q.04-How to find length of string in java without using length() method?**

**ANS = 1.** Convert string to char array using toCharArray method

2.Iterate over char array and incrementing length variable.

**Code; -**

**class LenghtOfStringMain{**

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

**String helloWorld="This is hello world";**

**System.out.println("length of helloWorld string :"+getLengthOfStringWithCharArray(helloWorld));**

**}**

**public static int getLengthOfStringWithCharArray(String str)**

**{**

**int length=0;**

**char[] strCharArray=str.toCharArray();**

**for(char c:strCharArray)**

**{**

**length++;**

**}**

**return length;**

**}**

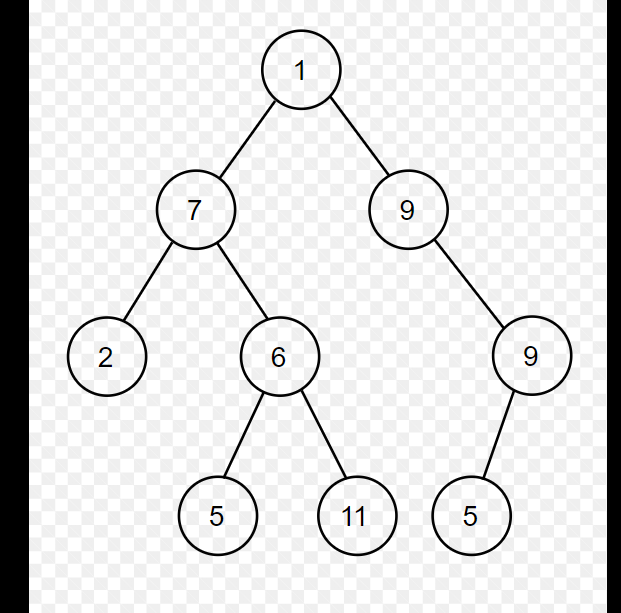
**}**

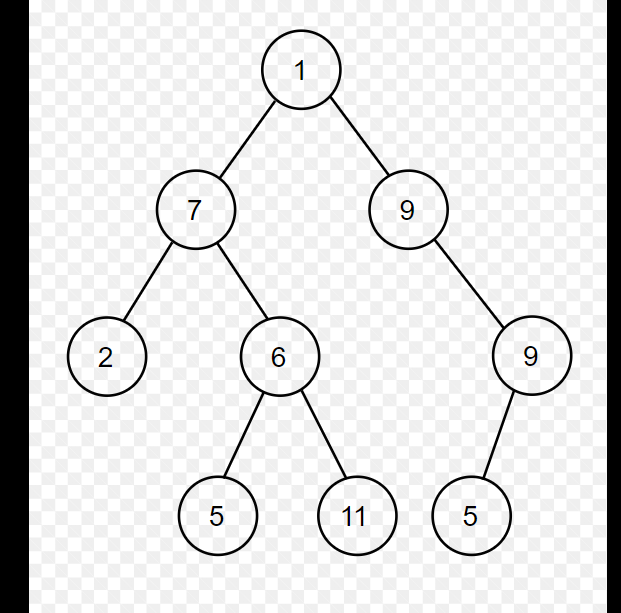
**Graphical user interface, text, application

Description automatically generated**

**Q.05: What is binary search tree?**

**ANS =** a binary search tree (BST), also called an ordered or sorted binary tree, is a rooted binary tree data structure with the key of each internal node being greater than all the keys in the respective node's left subtree and less than the ones in its right subtree.

****

****