JAVA

1.Write a Java program to create and throw custom exceptions?

```
class MyException extends Exception {
}
public class setText {
        public static void main(String args[])
        {
                try {
                        throw new MyException();
                }
                catch (MyException ex) {
                        System.out.println("Caught");
                        System.out.println(ex.getMessage());
                }
        }
}
```

2.How do we reverse a string?

```
class Revere string{
public static void main(String[] args)
{
string str = "ABCD";
String str2 ="     "
for (int i = str.length()-1; i>=0;i--)
{
Str2= St2 + str. charAt(i);
}
systern. Out.println (str2);
```

```
}
```

3.Write a program that detects the duplicate characters in a string.

```
public class Duplcharacter {
 public static void main(String argu[]) {

  String str = "prasanna123";
  int cnt = 0;
  char[] inp = str.toCharArray();
  System.out.println("Duplicate Characters are:");
  for (int i = 0; i < str.length(); i++) {
   for (int j = i + 1; j < str.length(); j++) {
    if (inp[i] == inp[j]) {
     System.out.println(inp[j]);
     cnt++;
     break;
    }
   }
  }
 }
}
```

4. Write a Program to remove duplicates in an ArrayList.

```
public class RemoveDuplicateArrayList {
    public static void main(String[] args) {
        List<String> l = new ArrayList<String>();
        l.add("Siva");
        l.add("Suriya");
        l.add("Siva");
```

```
            l.add("Abin");
            System.out.println(l.toString());
            Set<String> s = newLinkedHashSet<String>(l);
            System.out.println(s);
        }
    }
```

5.Write a program to demonstrate method overriding?

```
class Parent {
    void show()
    {
        System.out.println("Parent's show()");
    }
}
class Child extends Parent {
    void show()
    {
        System.out.println("Child's show()");
    }
}

class Main {
    public static void main(String[] args)
    {

        Parent obj1 = new Parent();
        obj1.show();
        Parent obj2 = new Child();
        obj2.show();
```

```
        }
}
```

6.How is an infinite loop declared in Java?

```
        import java.util.*;
        import java.lang.*;
        class Rextester
        {
            public static void main(String args[])
            {
          do
              {
               System.out.print("javaTpoint");
           System.out.print(" ");
              }while(true);
       }
        }
```

7.Write a program to demonstrate the method overloading by changing data types?

```
class Adder{
static int add(int a, int b){return a+b;}
static double add(double a, double b){return a+b;}
}
class TestOverloading2{
public static void main(String[] args){
System.out.println(Adder.add(11,11));
System.out.println(Adder.add(12.3,12.6));
}}
```

8.Write a program to demonstrate the method overloading by changing a number of arguments?

```
import java.io.*;
```

```
class Product {
	public int Prod(int a, int b, int c)
	{
		int prod1 = a * b * c;
		return prod1;
	}
	public double Prod(double a, double b, double c)
	{

		double prod2 = a * b * c;
		return prod2;
	}
}

class GFG {
	public static void main(String[] args)
	{

		Product obj = new Product();

		int prod1 = obj.Prod(1, 2, 3);
		System.out.println("Product of the three integer value :" + prod1);
		double prod2 = obj.Prod(1.0, 2.0, 3.0);
		System.out.println("Product of the three double value :" + prod2);
	}
}
```

9.How to read a file in Java?

```
import java.io.*;
public class FileReaderWithBufferedReader {
   public static void main(String[] args) throws IOException{We
      String file = "src/file.txt";
      BufferedReader bufferedReader = new BufferedReader(new FileReader(file));

      String curLine;
      while ((curLine = bufferedReader.readLine()) != null){
        System.out.println(curLine);
      }
      bufferedReader.close();
   }
}
```

10. What is getname in java with example?

```
import java.io.*;
public class solution {
        public static void main(String args[])
        {
                try {
                        File f = new File("c:\\users\\program.txt");
                        String Name = f.getName();
                        System.out.println("File Name : " + Name);
                }
                catch (Exception e) {
                        System.err.println(e.getMessage());
                }
        }
```

```
}
```

11.How to convert string to date in java in yyyy-mm-dd format?

```
import java.time.LocalDate;
import java.time.format.DateTimeFormatter;
import java.time.format.DateTimeParseException;
public class Main {
  public static void main(String[] args) {
    try {
      String string = "2023-04-14";
      LocalDate date = LocalDate.parse(string, DateTimeFormatter.ISO_DATE);
      System.out.println(date);
    }
    catch (IllegalArgumentException e) {
      System.out.println("Exception: " + e);
    }
    catch (DateTimeParseException e) {
      System.out.println("Exception: " + e);
    }
  }
}
```

12.Write a program to generate the following output in java?

```
public class Pattern
{
  public static void main(String args[])
  {
     int n = 5;
     for(int i=0;i<n;i++)
     {.
        for(int j=0;j<=i;j++)
        {
```

```
            System.out.print("* ");
          }
          System.out.println();
      }
  }
}
```

13. How to find duplicate characters in a string in Java?

```
public class Duplcharacter {
 public static void main(String argu[]) {

  String str = "prasanna123";
  int cnt = 0;
  char[] inp = str.toCharArray();
  System.out.println("Duplicate Characters are:");
  for (int i = 0; i < str.length(); i++) {
   for (int j = i + 1; j < str.length(); j++) {
    if (inp[i] == inp[j]) {
     System.out.println(inp[j]);
     cnt++;
     break;
    }
   }
  }
 }
}
```

14. How to remove special characters from a string in java?

```
public class RemoveSpecialCharacterExample1
{
```

```
public static void main(String args[])
{
String str= "This#string%contains^special*characters&.";
str = str.replaceAll("[^#&%*]", " ");
System.out.println(str);
}
}
```

15.How to remove duplicate elements from an array in Java?

```
public class RemoveDuplicateInArray{
   public static int removeDuplicateElements(int arr[], int n){
      if (n==0 || n==1){
         return n;
      }
      int j = 0;
      for (int i=0; i < n-1; i++){
         if (arr[i] != arr[i+1]){
            arr[j++] = arr[i];
         }
      }
      arr[j++] = arr[n-1];
      return j;
   }

   public static void main (String[] args) {
      int arr[] = {10,20,20,30,30,40,50,50};
      int length = arr.length;
      length = removeDuplicateElements(arr, length);
      for (int i=0; i<length; i++)
        System.out.print(arr[i]+" ");
   }
}
```

16. How to sort array in java?

```
import java.util.Arrays
```

```
public class
public static void main(String[] args) {
int[] ar= {15, 118, 35, 29, 174, 109, 21, 92, 1, 100};
Arrays.sort(ar);
System.out.printf("Modified ar[]: %s",)
Arrays.toString(ar));
}
```

17. Write a program to do bubble sort on an array in java.

```
public class BubbleSort {
   static void bubbleSort(int[] arr) {
      int n = arr.length;
      int temp = 0;
       for(int i=0; i < n; i++){
         for(int j=1; j < (n-i); j++){
                  if(arr[j-1] > arr[j]){
                        temp = arr[j-1];
                        arr[j-1] = arr[j];
                        arr[j] = temp;
                  }
            }
      }
  }
```

18. Write a Java program that sorts HashMap by value.

```
class Main {
  public static void main(String[] args) {
    LinkedHashMap<String, String> capitals = new LinkedHashMap();
    capitals.put("Nepal", "Kathmandu");
    Map<String, String> result = sortMap(capitals);
      for (Map.Entry entry : result.entrySet()) {
     System.out.print("Key: " + entry.getKey());
```

```
    System.out.println(" Value: " + entry.getValue());
  }
```

19. Write Java program that checks if two arrays contain the same elements.

```
public class sameelements {
 public static void main(String args[]) {
  int[] array1 = {};
  int[] array2 = { };
  System.out.println("The first array is: "+Arrays.toString(array1));
  System.out.println("The second array is: "+Arrays.toString(array2));
  Arrays.sort(array1);
  Arrays.sort(array2);
  System.out.println("The sorted first array is: "+Arrays.toString(array1));
  System.out.println("The sorted second array is: "+Arrays.toString(array2));
  System.out.println(" arrays contain the same elements? "
  +Arrays.equals(array1, array2));
}}
```

20. Write a Java program to show a NullPointerException.

```
Class NullPointerException
{
   public static void main (String[] args)
   {

      try
      {

         if (ptr.equals("a NullPointerException "))
            System.out.print("Same");
         else
            System.out.print("Not Same");
      }
      catch(NullPointerException )
      {
```

```
            System.out.print("NullPointerException Caught");
        }
    }
}
```

21. Show an example of switch expressions and multi-label case statements in Java.

```
 Publicclass
 {

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

                     static void print(Person person) {
                        String title = switch (person) {
                        };
                        System.out.printf(" ", person, title);
                     }
                  }
```

22. How do you write an interface with default and static method?

```
interface NewInterface {
        static void hello()
        {
                System.out.println("Hello, New Static ");
        }
        void overrideMethod(String str);
}
public class Interface implements NewInterface {
          public static void main(String[] args)
        {
                InterfaceDemo interfaceDemo = new InterfaceDemo();
                NewInterface.hello();
                interfaceDemo.overrideMethod("Hello, Override Method here");
```

```
        }
        public void overrideMethod(String str)
        {
                System.out.println(str);
        }
}
```

23. What is Stringjoiner in Java 8?

In java 8, a new class StringJoiner is introduced in the java.util package. Using this class we can join more than one strings with the specified delimiter, we can also provide prefix and suffix to the final string while joining multiple strings.

24. Write a Java program to print stars using for loop, where the number of stars printed should be equal to the row number?

```
public class Pattern
{
  public static void main(String args[])
  {
     int n = 5;
     for(int i=0;i<n;i++)
     {.
        for(int j=0;j<=i;j++)
        {
          System.out.print("* ");
        }
        System.out.println();
     }
  }
}
```

25.Write a Java program to demonstrate the usage of break and continue statements inside while loop?

```
public class Test {
   public static void main(String args[]) {
      int [] numbers = {10, 20, 30, 40, 50};
      for(int x : numbers ) {
         if( x == 30 ) {
            break;
         }
```

```
        System.out.print( x );
        System.out.print(" ");
      }
    }
}

public class Test {
   public static void main(String args[]) {
      int [] numbers = {10, 20, 30, 40, 50};
      for(int x : numbers ) {
         if( x == 30 ) {
            continue;
         }
         System.out.print( x );
         System.out.print(" ");
      }
   }
}
```

26. Write a Java Program to print the below output: * 1 * 12 * 123 * 1234 * 12345 * 123 456 * 1234567

```
class series
{
public static void main(String[] args)
{
 for(int i=1;i<=5;i++)
{
 for(int j=1;j<=i;j++)
 {
 System.out.print(j);
} System.out.println();
}
}
 }
```

27. Write a Java program to read and write a file?

```
public class  {

   public static void main(String args[]) throws IOException {
            FileReader in = null;
      FileWriter out = null;

      try {
```

```
      in = new FileReader("C:\\Raj\\Input-File.txt");
      out = new FileWriter("C:\\Raj\\Output-File.txt");

      int c;
      while ((c = in.read()) != -1) {
        out.write(c);
      }
      System.out.println("Reading and Writing in a file is done!!!");
    }
    catch(Exception e) {
         System.out.println(e);
    }
    finally {
      if (in != null) {
        in.close();
      }
      if (out != null) {
        out.close();
      }
    }
  }
}
```

28. Implement factorial using recursion

```
int main()
{
int i,fact=1,number;
printf("Enter a number: ");
scanf("%d",&number);
for(i=1;i<=number;i++){
    fact=fact*i;
 }
printf("Factorial of %d is: %d",number,fact);
return 0;
 }
```

29. Implement multiple inheritances using an interface

```
class ParentClass1
{
   void show()
```

```
   {
      System.out.println("ParentClass1");
   }
}

class ParentClass2
{
   void show()
   {
      System.out.println("ParentClass2");
   }
}
class SubClass extends ParentClass1, ParentClass2{
 public static void main(String[] args) {
  SubClass obj = new SubClass();

  obj.show();
 }
}
```

30. Java program for Enumeration

```
  class Enum{
public enum Season { WINTER, SPRING, SUMMER, FALL }
public static void main(String[] args) {
for (Season s : Season.values())
System.out.println(s);
}}
```

31. Implement method overloading & overriding in java

```
class Overloading{
static int add(int a,int b){return a+b;}
static int add(int a,int b,int c){return a+b+c;}
}

class Animal{
void eat(){System.out.println("eating...");}
}
```

```
class Dog extends Animal{
void eat(){System.out.println("eating bread...");}
}
```

32. Program to find duplicate values for ArrayList

```
public class DuplicateElement {
   public static void main(String[] args) {
      int [] arr = new int [] {1, 2, 3, 4, 2, 7, 8, 8, 3};
      System.out.println("Duplicate elements in given array: ");
      for(int i = 0; i < arr.length; i++) {
         for(int j = i + 1; j < arr.length; j++) {
            if(arr[i] == arr[j])
               System.out.println(arr[j]);
         }
      }
   }
}
```

33. Implement a program to merge two Arrays

```
import java.util.Arrays;
public class MergeTwoArrays{
        public static void main(String[] args)
        {
                int[] a = { 10, 20, 30, 40 };
                int[] b = { 50, 60, 70, 80 };
                int a1 = a.length;
                int b1 = b.length;
                int c1 = a1 + b1;
                int[] c = new int[c1];
                System.arraycopy(a, 0, c, 0, a1);
                System.arraycopy(b, 0, c, a1, b1);
                System.out.println(Arrays.toString(c));
        }
```

```
}
```

34. Implement a program to reverse a string using stack

```
import java.io.*;
import java.util.*;
class StringReverse {
   public static String ReverseString(String str)
   {
      char[] reverseString = new char[str.length()];
      Stack<Character> stack = new Stack<Character>();

      for (int i = 0; i < str.length(); i++) {
         stack.push(str.charAt(i));
      }
      int i = 0;
      while (!stack.isEmpty()) {
         reverseString[i++] = stack.pop();
      }

      return new String(reverseString);
   }
   public static void main(String[] args)
   {
      System.out.println(" Hello, World! ");
      Scanner sc = new Scanner(System.in);
      System.out.println(" Enter a String: ");
      String str = sc.next();
      System.out.println(" The entered String is: " + str);
      System.out.println(" The String in reverse order is: " + ReverseString(str));
   }
}
```

35. Implement a program to sort a map by value / Key

```
import java.util.*;
import java.util.Map.Entry;
```

```
class Main {

  public static void main(String[] args) {

    LinkedHashMap<String, String> capitals = new LinkedHashMap();
    capitals.put("Nepal", "Kathmandu");

    Map<String, String> result = sortMap(capitals);

    for (Map.Entry entry : result.entrySet()) {
      System.out.print("Key: " + entry.getKey());
      System.out.println(" Value: " + entry.getValue());
    }
  }

  public static LinkedHashMap sortMap(LinkedHashMap map) {
    List <Entry<String, String>> capitalList = new LinkedList<>(map.entrySet());

    Collections.sort(capitalList, (l1, l2) -> l1.getValue().compareTo(l2.getValue()));

    LinkedHashMap<String, String> result = new LinkedHashMap();

    for (Map.Entry<String, String> entry : capitalList) {
      result.put(entry.getKey(), entry.getValue());
    }

    return result;
  }
}
```

36. Write a Java Program for Fibonacci series.

```
Class Fibonacci{
public static void main(string args[])
{
int n=7;
int f=1, s=1, t;
System.out.println(f);
System out. println(s);
for(int i=3;i<=n; i++)
{
   t = f+s;
```

```
  System.out.println(t);
  f=s;
  s=t;
}
}
```

38. Compare StringBuffer with a string

```
public class {
        public static void main(String[] args) {
                String str = "apple";
                StringBuffer stringBuffer = new StringBuffer();
                stringBuffer.append('a');
                stringBuffer.append('p');
                stringBuffer.append('p');
                stringBuffer.append('l');
                stringBuffer.append('e');

                boolean result = str.contentEquals(stringBuffer);
                System.out.println("str and stringBuffer have same ? " + result);
        }
}
```

38. Constructor Overloading

```
class constructor{
     int  value1;
     int  value2;
     Demo(){
        value1 = 10;
        value2 = 20;
        System.out.println("Inside Constructor");
    }

    public void display(){
       System.out.println("Value1 === "+value1);
       System.out.println("Value2 === "+value2);
   }

  public static void main(String args[]){
      Demo d1 = new Demo();
     d1.display();
  }
```

```
}
```

39. Iterate the map values using lambda expression

```
import java.util.HashMap;
import java.util.Map;
public class IterateMapUsingLambda {
	public static void main(String[] args) {
		Map<String, Integer> prices = new HashMap<>();
		prices.put("Apple", 50);
		prices.put("Orange", 20);
		prices.put("Banana", 10);
		prices.put("Grapes", 40);
		prices.put("Papaya", 50);
		System.out.println("Fruit: " + k + ", Price: " + v));

	}
}
```

40. remove duplicates from sorted array

```
class Main
{
	  static int removeDuplicates(int arr[], int a)
	  {
		if (a==0 || a==1)
			return a;

		int[] temp = new int[a];

		int q = 0;
		for (int p=0; p<a-1; p++)
			if (arr[p] != arr[p+1])
				temp[q++] = arr[p];

		temp[q++] = arr[a-1];

		for (int p=0; p<q; p++)
			arr[p] = temp[p];

		return q;
	  }
```

```
        public static void main (String[] args)
        {
                int arr[] = {2,3,4,4,5,6,6,6,7,8,8};
                int a = arr.length;

                a = removeDuplicates(arr, a);
                System.out.print(a);
        }
}
```

41.swap two numbers without using temporary variable.

```
  class numbers {
        public static void main(String a[])
        {
              int x = 10;
               int y = 5;
               x = x + y;
               y = x - y;
               x = x - y;
               System.out.println("After swapping:"+ " x = " + x + ", y = " + y);
        }
}
```

42. Implement Exception Handling without Catch block.

```
      public class TryBlockWithouCatch{
  public static void main(String[] args) {
    try {
      System.out.println("Try Block");
    } finally {
      System.out.println("Finally Block");
    }
```

```
  }
}
```

43. Iterate the map values using lambda expression

```
import java.util.*;
public class MapIterateLambdaTest {
  public static void main(String[] args) {
    Map<String, Integer> ranks = new HashMap<String, Integer>();
    ranks.put("India", 1);
    ranks.put("Australia", 2);
    ranks.put("England", 3);
    ranks.put("Newzealand", 4);
    ranks.put("South Africa", 5);
    forEach using Lambda Expression
    ranks.forEach((k,v) -> System.out.println("Team : " + k + ", Rank : " + v));
  }
}
```

44. Find duplicate elements in a string

```
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]);
    }
  }
}
```

45. Iterate the map values using lambda expression

```
import java.util.*;
public class MapIterateLambdaTest {
  public static void main(String[] args) {
    Map<String, Integer> ranks = new HashMap<String, Integer>();
    ranks.put("India", 1);
    ranks.put("Australia", 2);
    ranks.put("England", 3);
    ranks.put("Newzealand", 4);
    ranks.put("South Africa", 5);
    forEach using Lambda Expression
    ranks.forEach((k,v) -> System.out.println("Team : " + k + ", Rank : " + v));
  }
}
```

46. Iterate a Linkedlist using Lambda's Expression

```
import java.util.*;
public class ListIterateLambdaTest {
  public static void main(String[] argv) {
```

```
    LinkedList<String> countryNames = new ArrayList<String>();
    countryNames.add("India");
    countryNames.add("England");
    countryNames.add("Australia");
    countryNames.add("Newzealand");
    countryNames.add("South Africa");
    countryNames.forEach(name -> System.out.println(name));
  }
}
```

47. Find Maximum repeated charcter count in a string

```
import java.util.*;
public class Main {
        static char getMaxOccurringChar(String str) {
                HashMap<Character, Integer> mp = new HashMap<>();
                    int n = str.length();
                char ans = 0;
                int cnt = 0;
                for(int i = 0; i < n; i++) {
                        char c = str.charAt(i);
                        mp.put(c, mp.getOrDefault(c, 0) + 1);
                        if(cnt < mp.get(c)) {
                                ans = c;
                                cnt = mp.get(c);
                        }
                }
                return ans;
        }
```

```
        public static void main(String[] args) {
                String str = "sample string";
                System.out.println("Max occurring character is: " +
getMaxOccurringChar(str));
        }
}
```

48. Implement quick sorting

```
import java.util.Arrays;
class Quicksort {
  static int partition(int array[], int low, int high) {
    int pivot = array[high];
    int i = (low - 1);
    for (int j = low; j < high; j++) {
      if (array[j] <= pivot) {
        i++;
        int temp = array[i];
        array[i] = array[j];
        array[j] = temp;
      }
    }
    int temp = array[i + 1];
    array[i + 1] = array[high];
    array[high] = temp;
    return (i + 1);
  }
  static void quickSort(int array[], int low, int high) {
    if (low < high) {
```

```
      int pi = partition(array, low, high);
      quickSort(array, low, pi - 1);
      quickSort(array, pi + 1, high);
    }
  }
}
class Main {
  public static void main(String args[]) {
    int[] data = { 8, 7, 2, 1, 0, 9, 6 };
    System.out.println("Unsorted Array");
    System.out.println(Arrays.toString(data));
    int size = data.length;
    Quicksort.quickSort(data, 0, size - 1);
    System.out.println("Sorted Array in Ascending Order ");
    System.out.println(Arrays.toString(data));
  }
}
```

49. Write a Java Program to iterate HashMap using While and advance for loop

```
import java.util.HashMap;
import java.util.Map;
public class GFG {
        public static void main(String[] args)
        {
                Map<String, String> foodTable= new HashMap<String, String>();
                foodTable.put("A", "Angular");
                foodTable.put("J", "Java");
                foodTable.put("P", "Python");
```

```
            foodTable.put("H", "Hibernate");
            for (Map.Entry<String, String> set :
                foodTable.entrySet()) {
                System.out.println(set.getKey() + " = "+ set.getValue());
            }
        }
}
```

50. Implement Merge sorting

```
class MergeSort {
    void merge(int arr[], int l, int m, int r)
    {
        int n1 = m - l + 1;
        int n2 = r - m;
        int L[] = new int[n1];
        int R[] = new int[n2];
        for (int i = 0; i < n1; ++i)
            L[i] = arr[l + i];
        for (int j = 0; j < n2; ++j)
            R[j] = arr[m + 1 + j];
        int i = 0, j = 0;
        int k = l;
        while (i < n1 && j < n2) {
            if (L[i] <= R[j]) {
                arr[k] = L[i];
                i++;
            }
            else {
```

```
                arr[k] = R[j];
                j++;
            }
            k++;
        }
        while (i < n1) {
            arr[k] = L[i];
            i++;
            k++;
        }
        while (j < n2) {
            arr[k] = R[j];
            j++;
            k++;
        }
    }
    void sort(int arr[], int l, int r)
    {
        if (l < r) {
            int m = l + (r - l) / 2;
            sort(arr, l, m);
            sort(arr, m + 1, r);
            merge(arr, l, m, r);
        }
    }
    static void printArray(int arr[])
    {
```

```
        int n = arr.length;
        for (int i = 0; i < n; ++i)
            System.out.print(arr[i] + " ");
        System.out.println();
    }
    public static void main(String args[])
    {
        int arr[] = { 12, 11, 13, 5, 6, 7 };

        System.out.println("Given Array");
        printArray(arr);
        MergeSort ob = new MergeSort();
        ob.sort(arr, 0, arr.length - 1);
        System.out.println("\nSorted array");
        printArray(arr);
    }
}
```

51. Java program for Enumeration

```
class EnumExample1{
public enum Season { WINTER, SPRING, SUMMER, FALL }
public static void main(String[] args) {
for (Season s : Season.values())
System.out.println(s);
}
}
```

52. compare two arrays and return the common elements

```
import java.io.*;
```

```
import java.util.*;
class GFG {
	private static void FindCommonElemet(String[] arr1,String[] arr2)
	{
		Set<String> set = new HashSet<>();
		for (int i = 0; i < arr1.length; i++) {
			for (int j = 0; j < arr2.length; j++) {
				if (arr1[i] == arr2[j]) {
					set.add(arr1[i]);
					break;
				}
			}
		}
		for (String i : set) {
			System.out.print(i + " ");
		}
	}
	public static void main(String[] args)
	{
		String[] arr1= { "Article", "in", "Geeks", "for", "Geeks" };
		String[] arr2 = { "Geeks", "for", "Geeks" };
		System.out.println("Array 1: "+ Arrays.toString(arr1));
		System.out.println("Array 2: "+ Arrays.toString(arr2));
		System.out.print("Common Elements: ");
		FindCommonElemet(arr1, arr2);
	}
}
```

53. Write a Java Program to find whether a string or number is palindrome or not.

```
class Main {
  public static void main(String[] args) {
    String str = "Radar", reverseStr = "";
    int strLength = str.length();
    for (int i = (strLength - 1); i >=0; --i) {
      reverseStr = reverseStr + str.charAt(i);
    }
    if (str.toLowerCase().equals(reverseStr.toLowerCase())) {
      System.out.println(str + " is a Palindrome String.");
    }
    else {
      System.out.println(str + " is not a Palindrome String.");
    }
  }
}
```

54. Implement more than one interface in a single class

```
interface Interface1
 {
    void f1();
 }
  interface Interface2
 {
     void f2();
 }
 class X implements Interface1,Interface2
 {
```

```
    public void f1()
    {
        System.out.println("Contens of Method f1()in Interface1");
     }
    public void f2()
    {
         System.out.println("Contens of Method f2()in Interface2") ;
    }
    public void f3()
    {
         System.out.println("Contens of Method f3() of Class X");
     }
}
   class MultipleInterface
{
    public static void main(String[] args)
   {
      Interface1 vl ;
      vl = new X();
     vl.f1();
      Interface2 v2;
      v2 = new X();
      v2.f2();
      X xl=new X();
      xl.f3();
   }
}
```

55. Iterate the LinkedHashMap values

```
import java.util.Iterator;
import java.util.LinkedHashMap;
import java.util.Set;
public class GFG {
	public static void main(String[] args)
	{

	LinkedHashMap<String,String>linkedHashMap=newLinkedHashMap<String,Stri
ng>();
		linkedHashMap.put("One", "First element");
		linkedHashMap.put("Two", "Second element");
		linkedHashMap.put("Three", "Third element");
		Set entrySet = linkedHashMap.entrySet();
		Iterator it = entrySet.iterator();
		System.out.println("LinkedHashMap entries : ");
		while (it.hasNext())
			System.out.println(it.next());
	}
}
```

56. Implement a program for encapsulation

```
class Person {
	private String name;
	private int age;
	public String getName() { return name; }

	public void setName(String name) { this.name = name; }
```

```
    public int getAge() { return age; }
    public void setAge(int age) { this.age = age; }
}
public class Main {
    public static void main(String[] args)
    {
        Person person = new Person();
        person.setName("John");
        person.setAge(30);
        System.out.println("Name: " + person.getName());
        System.out.println("Age: " + person.getAge());
    }
}
```

57. convert string to char and vice versa

```
public class StringToCharExample3{
public static void main(String args[]){
String s1="hello";
char[] ch=s1.toCharArray();
for(int i=0;i<ch.length;i++){
System.out.println("char at "+i+" index is: "+ch[i]);
}
}
}
```

58. Iterate the LinkedHashMap values

```
import java.util.Iterator;
import java.util.LinkedHashMap;
import java.util.Set;
```

```
public class GFG {
	public static void main(String[] args)
	{

	LinkedHashMap<String,String>linkedHashMap=newLinkedHashMap<String,Stri
ng>();
		linkedHashMap.put("One", "First element");
		linkedHashMap.put("Two", "Second element");
		linkedHashMap.put("Three", "Third element");
		Set entrySet = linkedHashMap.entrySet();
		Iterator it = entrySet.iterator();
		System.out.println("LinkedHashMap entries : ");
		while (it.hasNext())
			System.out.println(it.next());
	}
}
```

59. Implement a program for abstraction

```
abstract class Shape {
	String color;
	abstract double area();
	public abstract String toString();
	public Shape(String color)
	{
		System.out.println("Shape constructor called");
		this.color = color;
	}
	public String getColor() { return color; }
```

```
}
class Circle extends Shape {
	double radius;
	public Circle(String color, double radius)
	{
		super(color);
		System.out.println("Circle constructor called");
		this.radius = radius;
	}
	@Override double area()
	{
		return Math.PI * Math.pow(radius, 2);
	}
	@Override public String toString()
	{
		return "Circle color is " + super.getColor()
			+ "and area is : " + area();
	}
}
class Rectangle extends Shape {
	double length;
	double width;
	public Rectangle(String color, double length,double width)
	{
		super(color);
		System.out.println("Rectangle constructor called");
		this.length = length;
```

```
            this.width = width;
        }

        @Override double area() { return length * width; }

        @Override public String toString()
        {
            return "Rectangle color is " + super.getColor()
                    + "and area is : " + area();
        }
}
public class Test {
        public static void main(String[] args)
        {
            Shape s1 = new Circle("Red", 2.2);
            Shape s2 = new Rectangle("Yellow", 2, 4);
            System.out.println(s1.toString());
            System.out.println(s2.toString());
        }
}
```

60. Implement a program to handle more than one exception

```
class Main {
  public static void main(String[] args) {
    try {
      int array[] = new int[10];
      array[10] = 30 / 0;
    } catch (ArithmeticException e) {
```

```
    System.out.println(e.getMessage());
   } catch (ArrayIndexOutOfBoundsException e) {
    System.out.println(e.getMessage());
   }
  }
}
```

61. Implement bubble sorting

```
import java.util.*;
class BubbleSort {
        void bubbleSort(int arr[])
        {
                int n = arr.length;
                for (int i = 0; i < n - 1; i++)
                        for (int j = 0; j < n - i - 1; j++)
                                if (arr[j] > arr[j + 1]) {
                                        int temp = arr[j];
                                        arr[j] = arr[j + 1];
                                        arr[j + 1] = temp;
                                }
        }
        void printArray(int arr[])
        {
                int n = arr.length;
                for (int i = 0; i < n; ++i)
                        System.out.print(arr[i] + " ");
                System.out.println();
        }
```

```
    public static void main(String args[])
    {
        BubbleSort ob = new BubbleSort();
        int arr[] = { 5, 1, 4, 2, 8 };
        ob.bubbleSort(arr);
        System.out.println("Sorted array");
        ob.printArray(arr);
    }
}
```

62. Convert arraylist into string

```
import java.util.ArrayList;
import java.util.Arrays;
class GFG {
    public static void main(String[] args)
    {
        ArrayList<String> al = new ArrayList<String>();
        al.add("Anshul Aggarwal");
        al.add("Mayank Solanki");
        al.add("Abhishek Kelenia");
        al.add("Vivek Gupta");

        String[] str = new String[al.size()];
        for (int i = 0; i < al.size(); i++) {
            str[i] = al.get(i);
        }
        for (String k : str) {
            System.out.println(k);
```

```
                }
        }
}
```

63. Convert a set to stream

```
import java.util.*;
import java.util.stream.Stream;
class GFG {
        public static void main(String[] args) {
        Set<Integer> set = new HashSet<>();
        set.add(2);
        set.add(4);
        set.add(6);
        set.add(8);
        set.add(10);
        set.add(12);
        Stream<Integer> stream = set.stream();
        stream.forEach(elem->System.out.print(elem+" "));

        }
}
```

64. Nested If Else clause in java

```
public class code{
   public static void main(String[] args) {
      int n=24;
      if (n % 2 == 0){
         System.out.print("Even ");
         if (n % 6 == 0) {
```

```
            System.out.println("and divisible by 6");
        } else {
            System.out.println("and not divisible by 6");
        }
    }
else {
        System.out.println("Odd ");
        if(n % 3 == 0) {
            System.out.println("and divisible by 3");
        } else {
  System.out.println("and not divisible by 3");
        }
    }
  }
}
```

65. How to check Odd and Even Number in java.

```
import java.util.Scanner;
public class EvenOdd {
   public static void main(String[] args) {
      Scanner reader = new Scanner(System.in);
      System.out.print("Enter a number: ");
      int num = reader.nextInt();
      if(num % 2 == 0)
         System.out.println(num + " is even");
      else
         System.out.println(num + " is odd");
   }
```

```
}
```

66. Print Multiplication table Program in java

```
public class MultiplicationTable {
    public static void main(String[] args) {
        int num = 5;
        for(int i = 1; i <= 10; ++i)
        {
            System.out.printf("%d * %d = %d \n", num, i, num * i);
        }
    }
}
```

67. Check no is Armstrong or not in java Program

```
public class Armstrong {
    public static void main(String[] args) {
        int number = 371, originalNumber, remainder, result = 0;
        originalNumber = number;
        while (originalNumber != 0)
        {
            remainder = originalNumber % 10;
            result += Math.pow(remainder, 3);
            originalNumber /= 10;
        }
        if(result == number)
            System.out.println(number + " is an Armstrong number.");
        else
            System.out.println(number + " is not an Armstrong number.");
    }
```

```
}
```

68. How to add two matrix in java Program

```
public static void main(String args[]){
int a[][]={{1,3,4},{2,4,3},{3,4,5}};
int b[][]={{1,3,4},{2,4,3},{1,2,4}};
int c[][]=new int[3][3];
for(int i=0;i<3;i++){
for(int j=0;j<3;j++){
c[i][j]=a[i][j]+b[i][j];
System.out.print(c[i][j]+" ");
}
System.out.println();//new line
}
}
}
```

69. How to multiply two matrix in java Program

```
public class MatrixMultiplicationExample{
public static void main(String args[]){
int a[][]={{1,1,1},{2,2,2},{3,3,3}};
int b[][]={{1,1,1},{2,2,2},{3,3,3}};
int c[][]=new int[3][3];
for(int i=0;i<3;i++){
for(int j=0;j<3;j++){
c[i][j]=0;
for(int k=0;k<3;k++)
{
c[i][j]+=a[i][k]*b[k][j];
```

```
}
System.out.print(c[i][j]+" "); }
System.out.println();//new line
}
}
}
```

70. How to replace string with another string in java Program

```
public class Example {
   public static void main( String args[] ) {
      String str = new String("Good Harry Good");
      System.out.println( "Initial String : "+ str);
      str = str.replaceAll( "Good" , "Bad" );
      System.out.println( "The String after substitution : "+str );
   }
}
```

71. How to create method in java Program

```
class Main {
  public int addNumbers(int a, int b) {
    int sum = a + b;
    return sum;
  }
  public static void main(String[] args) {
    int num1 = 25;
    int num2 = 15;
    Main obj = new Main();
    int result = obj.addNumbers(num1, num2);
    System.out.println("Sum is: " + result);
```

```
  }
}
```

72. Find Length, Concatenate and Replace String in Java Program

```
class StringMethods
{
public static void main(String args[])
{
int n;
String s = "Java programming", t = "", u = "";
System.out.println(s);
n = s.length();
System.out.println("Number of characters = " + n);
t = s.replace("Java", "C++");
System.out.println(s);
System.out.println(t);
u = s.concat(" is fun");
System.out.println(s);
System.out.println(u);
}
}
```

73. How to create constructor overloading in java Program

```
class Box
{
        double width, height, depth;
        Box(double w, double h, double d)
        {
                width = w;
```

```
        height = h;
        depth = d;
    }
    Box()
    {
        width = height = depth = 0;
    }
    Box(double len)
    {
        width = height = depth = len;
    }
    double volume()
    {
        return width * height * depth;
    }
}
```

74. How to print date and time in java Program

```
import java.text.*;
import java.util.*;
public class GFG {
    public static void main(String args[])
    {
        SimpleDateFormat formatDate = new SimpleDateFormat(
            "dd/MM/yyyy HH:mm:ss z");
        Date date = new Date();
        formatDate.setTimeZone(TimeZone.getTimeZone("IST"));
        System.out.println(formatDate.format(date));
```

```
	}
}
```

75. How to convert string to date in java program

```
import java.time.Instant;
import java.time.format.DateTimeParseException;
class GFG {
	public static Instant getDateFromString(String string)
	{
		Instant timestamp = null;
		timestamp = Instant.parse(string);
		return timestamp;
	}
	public static void main(String[] args)
	{
		String string = "2018-10-28T15:23:01Z";
		try {
			Instant timestamp = getDateFromString(string);
			System.out.println(timestamp);
		}
		catch (DateTimeParseException e) {
			System.out.println("Exception: " + e);
		}
	}
}
```

76. How to ArrayList using list interface program in java

```
import java.util.List;
import java.util.ArrayList;
```

```
class Main {
   public static void main(String[] args) {
      List<Integer> numbers = new ArrayList<>();
      numbers.add(1);
      numbers.add(2);
      numbers.add(3);
      System.out.println("List: " + numbers);
      int number = numbers.get(2);
      System.out.println("Accessed Element: " + number);
      int removedNumber = numbers.remove(1);
      System.out.println("Removed Element: " + removedNumber);
   }
}
```

77. How to create LinkedHashSet program in java

```
import java.util.LinkedHashSet;
public class GFG {
        public static void main(String[] args)
        {
                LinkedHashSet<String> linkedset= new LinkedHashSet<String>();
                linkedset.add("A");
                linkedset.add("B");
                linkedset.add("C");
                linkedset.add("D");
                linkedset.add("A");
                linkedset.add("E");
                System.out.println("Size of LinkedHashSet = "+ linkedset.size());
                System.out.println("Original LinkedHashSet:"+ linkedset);
```

```
            System.out.println("Removing D from LinkedHashSet: "
                                        + linkedset.remove("D"));

            System.out.println(
                    "Trying to Remove Z which is not "
                    + "present: " + linkedset.remove("Z"));
            System.out.println("Checking if A is present="
                                        + linkedset.contains("A"));
            System.out.println("Updated LinkedHashSet: "+ linkedset);
        }
}
```

78. How to create PriorityQueue program in java

```
import java.util.*;
class PriorityQueueDemo {
        public static void main(String args[])
        {
            PriorityQueue<Integer> pQueue = new PriorityQueue<Integer>();
            pQueue.add(10);
            pQueue.add(20);
            pQueue.add(15);
            System.out.println(pQueue.peek());
            System.out.println(pQueue.poll());
            System.out.println(pQueue.peek());
        }
}
```

79. How to create Method Overriding program in java

```
class Animal {
  public void displayInfo() {
    System.out.println("I am an animal.");
  }
}
class Dog extends Animal {
  @Override
  public void displayInfo() {
    System.out.println("I am a dog.");
  }
}
class Main {
  public static void main(String[] args) {
    Dog d1 = new Dog();
    d1.displayInfo();
  }
}
```

80. How to create Find Factorial No using Recursion Program in java

```
class FactorialExample{
 public static void main(String args[]){
  int i,fact=1;
  int number=5;
  for(i=1;i<=number;i++){
     fact=fact*i;
  }
  System.out.println("Factorial of "+number+" is: "+fact);
 }
```

}