DSA Questions:

1. First program:

class main1{

public static void main(String[] args){

int a = 10;

int b = 10;

int c = a+b;

System.out.println(c);

}

}

1. Second program:

class main2{

public static void main(String[] args){

int a = 10;

int b = 20;

int c = b - a;

System.out.println(c);

}

}

1. Third program:

import java.util.Scanner;

public class day2 {

    public static void main(String[] args){

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter your age");

        int age = sc.nextInt();

        if(age > 18){

            System.out.println("Eligible for vote");

        }

        else{

            System.out.println("Not eligible for vote");

        }

    }

}

1. Loops:

public class loops {

    public static void main(String[] args){

        //initialization,condition,incrementation/decrementation

        for(int i=1;i<=1000;i++){

            System.out.println(i);

        }

    }

}

5. Check no. is positive, negative or zero:

import java.util.Scanner;

public class checkno {

    public static void main(String[] args){

        Scanner sc = new Scanner(System.in);

        int n = sc.nextInt();

        if(n>0){

            System.out.println("number is positive");

        }

        else if(n == 0){

            System.out.println("number is zero");

        }

        else{

            System.out.println("number is negative");

        }

    }

}

6. Find the largest no. :

import java.util.Scanner;

public class largest {

    public static void main(String[] args){

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter your first no.");

        int num1 = sc.nextInt();

        System.out.println("Enter your second no.");

        int num2 = sc.nextInt();

        System.out.println("Enter your third no.");

        int num3 = sc.nextInt();

        if(num1 > num2 && num1 >num3){

            System.out.println("First no. is greatest");

        }

        else if(num2 > num1 && num2 > num3){

            System.out.println("Second no. is greatest");

        }

        else{

            System.out.println("Third no. is greatest");

        }

    }

}

7. Find the natural no. :

import java.util.Scanner;

public class FirstNaturalno {

    public static void main(String[] args){

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter the no.");

        int n = sc.nextInt();

        for(int i = 0;i<=n;i++){

            System.out.println(i);

        }

    }

}

8. Factorial of a no. :

import java.util.Scanner;

public class factorial {

    public static void main(String[] args){

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter your no.");

        int n = sc.nextInt();

        int fact = 1;

        while(n>0){

            fact\*=n;

            n--;

        }

        System.out.println(fact);

    }

}

9. check Leap year:

import java.util.Scanner;

public class leapyear {

    public static void main(String[] args){

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter a year");

        int year = sc.nextInt();

        if(year%4 == 0 ){

            if(year%100 == 0){

                if(year%400 == 0){

                    System.out.println("It's a leap year");

                }

                else{

                    System.out.println("It's not a leap year");

                }

            }

            else{

                System.out.println("It's a leap year");

            }

        }

    }

}

10. No. is divisible by 3 & 5:

import java.util.Scanner;

public class divisible{

    public static void main(String[] args){

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter the number");

        int n = sc.nextInt();

        if(n%3 == 0){

            System.out.println("No. is divisible by 3");

        }

        else if(n%5 == 0){

            System.out.println("No. is devisible by 5");

        }

        else{

            System.out.println("No. is not devisible by 3 & 5");

        }

    }

}

11. Sum of Digits of a given no. :

import java.util.Scanner;

public class sumOfDigits {

    public static void main(String[] args){

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter a number");

        int n = sc.nextInt();

        int rm = 0;

        while(n>0){

            int l = n % 10;

            rm = rm + l;

            n = n/10;

        }

        System.out.println(rm);

    }

}

12. Find the even no’s :

import java.util.Scanner;

public class printEven {

    public static void main(String[] args){

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter the range");

        int n = sc.nextInt();

        for(int i=0;i<=n;i+=2){

            System.out.println(i);

        }

    }

}

13. Check the given no. is palindrome or not :

public class palindrome {

    public static void main(String[] args) {

        int n = 1231;

        int new1 = n;

        int l;

        int rm = 0;

        while(n > 0) {

            l = n % 10;

            rm = rm \* 10 + l;

            n = n / 10;

        }

        if(rm == new1) {

            System.out.println(true);

        } else {

            System.out.println(false);

        }

    }

}

14. Reverse a no. :

import java.util.Scanner;

public class reverseno {

    public static void main(String[] args){

        Scanner sc = new Scanner(System.in);

        System.out.println("enter the number you want to reverse");

        int n = sc.nextInt();

        int rm = 0;

        while(n>0){

            int l = n%10;

            rm =  rm\*10 + l;

            n /=10;

        }

        System.out.println(rm);

    }

}

15. print a table :

import java.util.Scanner;

public class multiplication {

    public static void main(String[] args){

        Scanner sc = new Scanner(System.in);

        System.out.println("enter the no. for table");

        int n = sc.nextInt();

        for(int i =1;i<=10;i++){

            System.out.println(n + " \* "+ i + " = " + (i\*n) );

        }

    }

}

16. String Palindrome :

public class StringPalindrome {

    public static boolean isPalindrome(String str) {

        int left = 0;

        int right = str.length() - 1;

        while (left < right) {

            if (str.charAt(left) != str.charAt(right)) {

                return false;

            }

            left++;

            right--;

        }

        return true;

    }

    public static void main(String[] args) {

        String str = "mada";

        System.out.println(isPalindrome(str));

    }

}

17. Anagram :

import java.util.Arrays;

public class Anagram {

    public static boolean isAnagram(String str1, String str2){

        if(str1.length() != str2.length()){

            return false;

        }

        char []a = str1.toCharArray();

        char []b = str2.toCharArray();

        Arrays.sort(a);;

        Arrays.sort(b);

        return Arrays.equals(a, b);

    }

    public static void main(String[] args){

        String str1 = "listen";

        String str2 = "silents";

        System.out.println(isAnagram(str1, str2));

    }

}

18. Count Vowels:

public class countVowel {

    public static int countVowels(String str){

        int count = 0;

        for(char ch: str.toLowerCase().toCharArray()){

            if("aeiou".indexOf(ch) != -1){

                count++;

            }

        }

        return count;

    }

    public static void main(String[] args){

        String str = "Hello World";

        System.out.println("number of vowels are " + countVowels(str));

    }

}

19. Reverse String:

public class revString {

    public static String reverse(String str){

        String srt = "";

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

            srt += str.charAt(i);

        }

        return srt;

    }

    public static void main(String[] args){

        String str = "Vikas";

        System.out.println(reverse(str));

    }

}

20. Find the first non repeating character:

public class nonRepeatedChar {

    public static char Repeated(String str){

        int [] count = new int[256];

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

            count[str.charAt(i)]++;

        }

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

            if(count[str.charAt(i)] == 1){

                return str.charAt(i);

            }

        }

        return ' ';

    }

    public static void main(String[] args){

        String str = "Gaurisha";

        System.out.println("the first non repeated character is " + Repeated(str));

    }

}

21.Count frequency of each character:

public class CharFrequency {

    public static void main(String[] args) {

        String input = "programming"; // example string

        countCharacters(input);

    }

    public static void countCharacters(String str) {

        int[] count = new int[256]; // ASCII size

        // Count frequency of each character

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

            count[str.charAt(i)]++;

        }

        // Display frequency of each character

        boolean[] printed = new boolean[256]; // To avoid printing duplicates

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

            char ch = str.charAt(i);

            if (!printed[ch]) {

                System.out.println(ch + " = " + count[ch]);

                printed[ch] = true;

            }

        }

    }

}

22. Check the given no. is digit or not :

public class OnlyDigitsCheck {

    public static void main(String[] args) {

        String input = "123456"; // change this to test other strings

        if (containsOnlyDigits(input)) {

            System.out.println("The string contains only digits.");

        } else {

            System.out.println("The string contains other characters too.");

        }

    }

    public static boolean containsOnlyDigits(String str) {

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

            if (!Character.isDigit(str.charAt(i))) {

                return false;

            }

        }

        return !str.isEmpty(); // returns false for empty string

    }

}

23. Title case :

public class TitleCaseExample {

    public static void main(String[] args) {

        String input = "hello world this is java";

        String result = convertToTitleCase(input);

        System.out.println("Title Case: " + result);

    }

    public static String convertToTitleCase(String str) {

        String[] words = str.split(" ");

        String result = "";

        for (String word : words) {

            if (word.length() > 0) {

                String firstLetter = word.substring(0, 1).toUpperCase();

                String remainingLetters = word.substring(1).toLowerCase();

                result += firstLetter + remainingLetters + " ";

            }

        }

        return result.trim(); // Remove the extra space at the end

    }

}

24. Replace Space:

public class ReplaceSpaces {

    public static void main(String[] args) {

        String str = "Hello world this is Java";

        String result = str.replace(' ', '-');

        System.out.println("Original: " + str);

        System.out.println("Modified: " + result);

    }

}

25. Array first program:

public class Arrays123 {

    public static void main(String[] args){

        int []arr = {1,2,3,4,5,6,7,8};

        for(int i=0;i<=arr.length - 1;i++){

            System.out.println(arr[i]);

        }

    }

}

26. Print an array:

public class printArray {

    public static void main(String[] args){

        String []arr = {"H","e","l","o","w","o","r"};

        for(int i=0;i<=arr.length - 1;i++){

            System.out.println(arr[i]);

        }

    }

}

27. Check the no. is present in array or not:

import java.util.Scanner;

public class CheckArray {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int[] arr = {1, 2, 3, 4, 5, 6, 7, 8, 9};

        System.out.println("Enter your value:");

        int n = sc.nextInt();

        boolean found = false;

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

            if (arr[i] == n) {

                found = true;

                break;

            }

        }

        if (found) {

            System.out.println(n + " is present in the array");

        } else {

            System.out.println(n + " is not present in the array");

        }

    }

}

28. Check the String array :

import java.util.Scanner;

public class checkStringArray {

    public static void main(String[] args){

        Scanner sc = new Scanner(System.in);

        String []arr = {"Vikas","Neeraj","java","Vansh"};

        System.out.println("Enter the string");

        String n = sc.next();

        boolean found = false;

        for(int i = 0;i<=arr.length - 1;i++){

            if(arr[i].equals(n)){

                found = true;

                break;

            }

        }

        if(found){

            System.out.println(n + " is present in the string");

        }

        else{

            System.out.println(n + " is not present in the string");

        }

    }

}

29. Sum of elements of an array:

public class SumOfArray {

    public static void main(String[] args){

        int []arr = {1,2,3,4,5,6,7,8,9};

        int sum = 0;

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

            sum += arr[i];

        }

        System.out.println(sum);

    }

}

30. Max element in array :

public class maxInArray {

    public static void main(String[] args) {

        int []arr = {1,2,3,4,5,67,7,8,9};

        int max = arr[0];

        for(int i=1;i<arr.length;i++){

            if(max < arr[i]){

                max = arr[i];

            }

        }

        System.out.println(max);

     }

}

31. Reverse an array :

public class ReverseArray {

    public static void main(String[] args) {

        int []arr = {1,2,3,4,5,6};

        System.out.println("you want a reverse array");

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

            System.out.println(arr[i]);

        }

    }

}

32. Check the array is sorted or not :

public class ArrayIsSorted {

    public static void main(String[] args){

        int []arr = {1,2,3,4,5,6,7,8,9};

        boolean found = true;

        for(int i=1;i<arr.length;i++){

            if(arr[i - 1] > arr[i] ){

                found = false;

                break;

            }

        }

        if(found){

            System.out.println("Array is sorted");

        }

        else{

            System.out.println("Array is not sorted");

        }

    }

}

33. Object class introduction:

public class ObjectClassIntro {

    public static void main(String[] args){

        //Vikas v = new Vikas();

        //Neeraj n = new Neeraj();

        // v.run();

        // n.sir();

        Vikas.run();

        Neeraj.sir();

    }

}

class Vikas{

   public static void run(){

    System.out.println("Hi Vikas");

   }

}

class Neeraj{

    public static void sir(){

        System.out.println("Neeraj is a teacher");

    }

}

34. Car class :

public class CarObject {

    public static void main(String[] args){

       Car c = new Car();

       c.brand();

       c.model();

       c.year();

    }

}

class Car{

    void model(){

        System.out.println("M5");

    }

    void brand(){

        System.out.println("BMW");

    }

    void year(){

        System.out.println("2023");

    }

}

35. Rectangle class:

public class rectangleClass {

    //Create a Rectangle class with width and height attributes.

    //Add a method to calculate the area. Create objects and find the area.

    private double w;

    private double h;

    public rectangleClass(double w, double h){

        this.w = w;

        this.h = h;

    }

    public double area(){

        return w\*h;

    }

    public static void main(String[] args) {

        rectangleClass r = new rectangleClass(4, 10);

        System.out.println(r.area());

    }

}

36. Student details:

public class student {

    private String name;

    private int rollnumber;

    private double marks;

    public student(String name, int rollnumber,double marks){

        this.name = name;

        this.rollnumber = rollnumber;

        this.marks = marks;

    }

    public void Display(){

        System.out.println("name of student is " + name);

        System.out.println("Roll number of student is " + rollnumber);

        System.out.println("Marks of student is " + marks);

    }

    public static void main(String[] args) {

        student s = new student("Vikas", 959, 99.9);

        s.Display();

    }

}

37.Inheritence first program:

public class inheritance1 {

    public static void main(String[] args) {

        Dog d = new Dog();

        d.makesound();

    }

}

class Animal{

    void makesound(){

        System.out.println("Animals can make sound");

    }

}

class Dog extends Animal{

    void makesound(){

        System.out.println("Dog's can bark");

    }

}

38. Inheritence 2nd program:

public class inheritence2 {

    public static void main(String[] args) {

        Student s = new Student();

    }

}

class person{

    public person(){

        System.out.println("person is alive");

    }

}

class Student extends person{

    //this is constructor when using public with same name as class.

    public Student(){

        super();

        System.out.println("Student is alive");

    }

}

39. Inheritence 3rd program:

public class inheritence3 {

    public static void main(String[] args) {

        me m = new me();

    }

}

class parent{

    //creating a constructor.

    public parent(){

        System.out.println("he is may father");

    }

}

class me extends parent{

    public me(){

        super();

        System.out.println("she is my mother");

    }

}

40.Inheritence 4th program:

public class inheritence4 {

    public static void main(String[] args){

        car c = new car();

        c.speed();

    }

    //Create a class Vehicle with a speed variable and a method displaySpeed().

    // Derive a class Car that has a speed variable too. Use super.speed and this.

    //speed to print both values.

}

class veh{

    int s = 100;

}

class car extends veh{

    int s = 200;

    void speed(){

        System.out.println("This is the speed of class veh " + this.s);

        System.out.println("This is the speed of class car " + super.s);

    }

}

41.Inheritence 5th Program:

public class inheritence5 {

    //Create a class LivingThing, then subclass Animal, and then subclass Bird.

    //Each class should have a method describe(). Call describe() from a Bird object.

    public static void main(String[] args){

        Bird b = new Bird();

        b.describe();

    }

}

class LivingThing{

    void describe(){

        System.out.println("this is living class");

    }

}

class Animal extends LivingThing{

    void describe(){

        System.out.println("this is Animal class");

    }

}

class Bird extends Animal{

    void describe(){

        System.out.println("this is bird class");

    }

}

42. Inheritence 6th program:

public class inheritence6 {

    //Create a class Shape with method draw().

    //Subclass it as Circle and Square,each overriding draw().

    //Use a Shape reference to call the method on both subclasses.

    public static void main(String[] args){

        shape s = new Circle();

        shape n = new Square();

        s.draw();

        n.draw();

    }

}

class shape{

    void draw(){

        System.out.println("this is shape class");

    }

}

class Circle extends shape{

    void draw(){

        System.out.println("this is Circle class");

    }

}

class Square extends shape{

    void draw(){

        System.out.println("this is Square class");

    }

}

43.Inheritence 7th program:

class inhert{

    public static void main(String[] args){

        lion l = new lion();

        l.hunt();

        l.flying();

        l.dog();

    }

}

class animal{

    void dog (){

        System.out.println("Bark");

    }

}

class bird extends animal{

    void flying(){

        System.out.println("flying");

    }

}

class lion extends bird{

    void hunt(){

        System.out.println("hunting");

    }

}

43. Polymorphism 1st program:

public class polymorphism1 {

public static void main(String[] args) {

    poly p = new poly();

    p.same();

    p.same(1);

    p.same(1,2);

    p.same("Vikas",10);

    p.same(1.99f,10);

}

}

class poly{

    void same(){

        System.out.println("this is first");

    }

    void same(int a){

        System.out.println("this is second");

    }

    void same(int a,int b){

        System.out.println("this is third");

    }

    void same(String c,int a){

        System.out.println("this is forth");

    }

    void same(float d,int a){

        System.out.println("this is fifth");

    }

}

44. Polymorphism 2nd Program:

public class polymorphism2 {

    public static void main(String[] args){

        maths m = new maths();

        m.print();

        m.print(959);

        m.print(10,20);

        m.print("Vikas","Verma");

        m.print(1.9f,1.1f);

    }

}

class maths{

    void print(){

        System.out.println("nothing is their");

    }

    void print(int a){

        System.out.println(a);

    }

    void print(int a,int b){

        System.out.println(a+b);

    }

    void print(String a,String b){

        System.out.println(a + b);

    }

    void print(float a , float b){

        System.out.println(a + b);

    }

}

45.polymorphism 3rd program:

public class polymorphism3 {

    public static void main(String[] args) {

        Verma v = new Verma();

        v.same();

    }

}

class Vikas{

    void same(){

        System.out.println("this first class");

    }

}

class Verma extends Vikas{

    void same(){

        System.out.println("This is second class");

    }

}

46. Polymorphism 3rd program:

public class polymorphism4 {

    public static void main(String[] args) {

        science s = new science();

        s.lesson();

    }

}

class maths{

    void lesson(){

        System.out.println("this is maths");

    }

}

class science extends maths{

    void lesson(){

        System.out.println("This is science");

    }

}

47.Polymorphism 5th program:

class polymorphism6{

    public static void main(String[] args){

        calculate c = new calculate();

        c.maths();

        c.maths(10);

        c.maths(2,10);

    }

}

class calculate{

    void maths(){

        System.out.println("Here is nothing");

    }

    void maths(int a){

        System.out.println("this is first integer " + a);

    }

    void maths(int a,int b){

        System.out.println("these are two variable " +a + " " + b);

    }

}

48. Polymorphism 6th program:

class polymorphism7{

    public static void main(String[] args){

        rollno a = new rollno();

        a.details();

    }

}

class student{

    void details(){

        System.out.println("My name is vikas");

    }

}

class address extends student{

    void details(){

        System.out.println("manimajra");

    }

}

class rollno extends address{

    void details(){

        System.out.println("2210990959");

    }

}

49.Polymorphism 7th program:

class polym{

    public static void main(String[] args){

        overload o = new overload();

        o.maths();

        o.maths(2);

        o.maths("Vikas");

    }

}

class overload{

    void maths(){

        System.out.println("nothing");

    }

    void maths(int a){

        System.out.println(a + " is present");

    }

    void maths(String a){

        System.out.println(a + " is String");

    }

}

50.Polymorphism 8th program:

class polym1{

    public static void main(String[] args){

        cars c = new cars();

        c.brand();

    }

}

class bike{

    void brand(){

        System.out.println("BMW");

    }

}

class cars extends bike{

    void brand(){

        System.out.println("Pagani");

    }

}

51.Encapsulation 1st program:

public class Encap1 {

    public static void main(String[] args){

        login l = new login();

        l.setUsername("Vikas");

        l.setPassword(9590);

        System.out.println("This is password "+ l.getPassword());

        System.out.println("This is username " + l.getUsername());

    }

}

class login{

    private String username;

    private int  password;

    public String getUsername() {

        return username;

    }

    public int getPassword() {

        return password;

    }

    public void setUsername(String username) {

        this.username = username;

    }

    public void setPassword(int password) {

        this.password = password;

    }

}

52. Encapsulation 2nd program:

class Encap3{

    public static void main(String[] args){

        Student s = new Student();

        s.setgrade("A++");

        s.setrollNo(9590);

        System.out.println("This is my rollNo. " + s.getrollNo());

        System.out.println("This is my grade " + s.getgrade());

    }

}

class Student{

    private int rollNo;

    private String grade;

    public int getrollNo(){

        return rollNo;

    }

    public String getgrade(){

        return grade;

    }

    public void setrollNo(int rollNo){

        this.rollNo = rollNo;

    }

    public void setgrade(String grade){

        this.grade =grade;

    }

}

53.Encapsulation 3rd program:

//Write a class Car with private fields model and speed.

//Add validation in the setter so that speed cannot be negative. Show its use in the main method.

class Encap4{

    public static void main(String[] args){

        car c = new car();

        c.setSpeed(-100);

        c.setmodel("BMW");

        System.out.println("This is model of car " + c.getmodel());

        System.out.println("This is the speed of car " + c.getSpeed());

    }

}

class car {

    private String model;

    private int Speed;

    public String getmodel(){

        return model;

    }

    public int getSpeed(){

        return Speed;

    }

    public void setmodel(String model){

        this.model = model;

    }

    public void setSpeed(int Speed){

        if(Speed > 0){

            this.Speed = Speed;

        }

        else{

            this.Speed = 0;

        }

    }

}

54. Encapsulation 4th program:

//Create a class Book with private fields title, author, and price.

//Implement getters and setters for each, and create a method displayDetails() that prints all book information.

class Encap5{

    public static void main(String[] args){

        Book b = new Book();

        b.settitle("Ramayan");

        b.setauthor("Hnauman");

        b.setprice(108);

        b.DisplayDetails();

    }

}

class Book{

    private String title;

    private String author;

    private int price;

    public String gettitle(){

        return title;

    }

    public String getauthor(){

        return author;

    }

    public int getprice(){

        return price;

    }

    public void settitle(String title){

        this.title = title;

    }

    public void setauthor(String author){

        this.author = author;

    }

    public void setprice(int price){

        this.price = price;

    }

    public void DisplayDetails(){

        System.out.println("this is title " + gettitle());

        System.out.println("this is Author's name " + getauthor());

        System.out.println("this is price " + getprice());

    }

}

55. Encapsulation 4th program:

//Create a class Book with private fields title, author, and price.

//Implement getters and setters for each, and create a method displayDetails() that prints all book information.

class Encap5{

    public static void main(String[] args){

        Book b = new Book();

        b.settitle("Ramayan");

        b.setauthor("Hnauman");

        b.setprice(108);

        b.DisplayDetails();

    }

}

class Book{

    private String title;

    private String author;

    private int price;

    public String gettitle(){

        return title;

    }

    public String getauthor(){

        return author;

    }

    public int getprice(){

        return price;

    }

    public void settitle(String title){

        this.title = title;

    }

    public void setauthor(String author){

        this.author = author;

    }

    public void setprice(int price){

        this.price = price;

    }

    public void DisplayDetails(){

        System.out.println("this is title " + gettitle());

        System.out.println("this is Author's name " + getauthor());

        System.out.println("this is price " + getprice());

    }

}

56.Encapsulation 5th program:

class Encap6{

     public static void main(String[] args) {

        Employee emp = new Employee(101, 50000);

        emp.displayInfo();

        emp.applyBonus(5000);

        System.out.println("After applying bonus:");

        emp.displayInfo();

    }

}

class Employee {

    private int id;

    private double salary;

    public int getId() {

        return id;

    }

    public void setId(int id) {

        this.id = id;

    }

    public double getSalary() {

        return salary;

    }

    public void setSalary(double salary) {

        this.salary = salary;

    }

    public void applyBonus(double bonusAmount) {

        this.salary += bonusAmount;

    }

    public void displayInfo() {

        System.out.println("Employee ID: " + id);

        System.out.println("Salary: ₹" + salary);

    }

}

57.Encapsulation 6th program:

class Encap7{

    public static void main(String[] args) {

        Temperature temp = new Temperature();

        temp.setTemperature(25);

        double fahrenheit = temp.toFahrenheit();

        System.out.println("Temperature in Fahrenheit: " + fahrenheit);

    }

}

class Temperature {

    private double celsius;

    public void setTemperature(double celsius) {

        this.celsius = celsius;

    }

    public double toFahrenheit() {

        return (celsius \* 9/5) + 32;

    }

}

58.Encapsulation 7th program:

class Encap8{

    public static void main(String[] args) {

        Rectangle rect = new Rectangle();

        rect.setLength(5);

        rect.setWidth(10);

        double area = rect.calculateArea();

        System.out.println("Area of rectangle: " + area);

        rect.setLength(-5);  // Invalid

        rect.setWidth(-10);   // Invalid

    }

}

class Rectangle {

    private double length;

    private double width;

    public double getlength(){

        return length;

    }

    public double getwidth(){

        return width;

    }

    public void setLength(double length) {

        if (length > 0) {

            this.length = length;

        } else {

            System.out.println("Length must be positive.");

        }

    }

    public void setWidth(double width) {

        if (width > 0) {

            this.width = width;

        } else {

            System.out.println("Width must be positive.");

        }

    }

    public double calculateArea() {

        return length \* width;

    }

}

60. Encapsulation 8th program:

class Encap9{

     public static void main(String[] args) {

        Login login = new Login();

        login.setUsername("user123");

        login.setPassword("mypassword");

        System.out.println("Username: " + login.getUsername());

        System.out.println("Password: " + login.getPassword());

        login.setPassword("short");

    }

}

class Login {

    private String username;

    private String password;

    public String getUsername() {

        return username;

    }

    public String getPassword() {

        return password;

    }

    public void setUsername(String username) {

        this.username = username;

    }

    public void setPassword(String password) {

        if (password.length() > 6) {

            this.password = password;

        } else {

            System.out.println("Password must be more than 6 characters.");

        }

    }

}

61.Encapsulation 9th program:

class Encap10{

    public static void main(String[] args) {

        Circle circle = new Circle();

        circle.setRadius(5);

        System.out.println("Radius: " + circle.getRadius());

        System.out.println("Area of the circle: " + circle.getArea());

        circle.setRadius(-3);

    }

}

class Circle {

    private double radius;

    public void setRadius(double radius) {

        if (radius >= 0) {

            this.radius = radius;

        } else {

            System.out.println("Radius cannot be negative.");

        }

    }

    public double getRadius() {

        return radius;

    }

    public double getArea() {

        return Math.PI \* radius \* radius; // Formula: Area = π \* r^2

    }

}

62.Abstraction 1st program:

class Abst{

    public static void main(String[]args){

        abc a=new abc();

        a.sleep();

        a.ask();

    }

}

abstract class Vikas{

    void sleep(){

        System.out.println("Hi I am sleepping");

    }

    abstract void ask();

}

class abc extends Vikas{

    void ask(){

        System.out.println("I am asking for something");

    }

}

63.Abstraction 2nd program:

class Abst2{

    public static void main(String[] args){

        abc a = new abc();

        a.sleep();

        a.running();

        a.playing();

    }

}

interface Vikas{

    abstract void sleep();

    abstract void running();

    abstract void playing();

}

class abc implements Vikas{

    public void sleep(){

        System.out.println("I am sleeping");

    }

    public void running (){

        System.out.println("i am running");

    }

    public void playing (){

        System.out.println("i am playing");

    }

}

64.Abstraction 3rd program:

class Abst3{

    //Create an abstract class Shape with an abstract method area().

    //Derive classes Circle and Rectangle and implement the area() method.

    public static void main(String[] args){

        circle c = new circle();

        c.area();

        rectangle r = new rectangle();

        r.area();

    }

}

abstract class Shape{

    abstract void area();

}

class circle extends Shape{

    void area(){

        System.out.println("this is circle's area");

    }

}

class rectangle extends Shape{

    void area(){

        System.out.println("This is rectangle's area");

    }

}

65. Abstraction 4th program:

class Abst4{

    //Create an abstract class Animal with an abstract method makeSound().

    //Extend it to Dog and Cat, each with their own implementation of makeSound().

    public static void main(String[] args){

        Dog d = new Dog();

        Cat c = new Cat();

        d.makeSound();

        c.makeSound();

    }

}

interface Animal{

    abstract void makeSound();

}

class Dog implements Animal{

    public void makeSound(){

        System.out.println("Hi! I am Dog");

    }

}

class Cat implements Animal{

    public void makeSound(){

        System.out.println("Hi! I am car");

    }

}

66. Abstraction 5th program:

class Abst5{

    //Define an abstract class Employee with abstract method calculateSalary().

    //Implement this method in FullTimeEmployee and PartTimeEmployee.

    public static void main(String[] args){

        FullTimeEmployee f = new FullTimeEmployee();

        //PartTimeEmployee p = new PartTimeEmployee(3000);

        f.calculateSalary(3);

        //p.calculateSalary();

    }

}

interface Employee{

    abstract void calculateSalary(int a);

}

class FullTimeEmployee implements Employee{

    public void calculateSalary(int a){

        System.out.println( a\*12 + " this is your salary");

    }

}

67. Abstraction 6th program:

class Abst6{

    //Create an abstract class Vehicle with an abstract method move().

    //Implement it in classes Car and Bike.

    public static void main(String[] args){

        Car c = new Car();

        Bike b = new Bike();

        c.move();

        b.move();

    }

}

interface Vehicle {

    abstract void move();

}

class Car implements Vehicle{

    public void move(){

        System.out.println("this is a car");

    }

}

class Bike implements Vehicle{

    public void move(){

        System.out.println("This is a bike");

    }

}

68. Abstraction 7th program:

class Abst7{

    //Define an abstract class Bank with method getRateOfInterest().

    //Extend it to SBI, ICICI, and HDFC with their respective interest rates.

    public static void main(String[] args){

        SBI s = new SBI();

        ICICI i = new ICICI();

        HDFC h = new HDFC();

        s.getRateOfInterest();

        i.getRateOfInterest();

        h.getRateOfInterest();

    }

}

interface Bank{

    abstract void getRateOfInterest();

}

class SBI implements Bank{

    public void getRateOfInterest(){

        System.out.println("1000");

    }

}

class ICICI implements Bank{

    public void getRateOfInterest(){

        System.out.println("2000");

    }

}

class HDFC implements Bank{

    public void getRateOfInterest(){

        System.out.println("3000");

    }

}

69.Abstraction 8th program:

class Abst8{

    //Create an abstract class Payment with an abstract method pay().

    //Extend it to classes like CreditCard and UPIPayment.

    public static void main(String[] args){

        CreditCard c = new CreditCard();

        UPIPayment u = new UPIPayment();

        c.pay();

        u.pay();

    }

}

abstract class Payment{

    abstract void pay();

}

class CreditCard extends Payment{

    void pay(){

        System.out.println("this is credit card");

    }

}

class UPIPayment extends Payment{

    void pay(){

        System.out.println("This is upi payment");

    }

}

70.Abstraction 9th program:

class Abst9{

    //Create an abstract class Appliance with method turnOn() and turnOff().

    //Extend it to Fan and Light

    public static void main(String[] args){

        Fan f = new Fan();

        Light l = new Light();

        f.turnOff();

        f.turnOn();

        l.turnOff();

        l.turnOn();

    }

}

abstract class Appliance{

    abstract void turnOn();

    abstract void turnOff();

}

class Fan extends Appliance{

    void turnOn(){

        System.out.println("this is running fan");

    }

    void turnOff(){

        System.out.println("this is still fan");

    }

}

class Light extends Appliance{

    void turnOn(){

        System.out.println("This light is on");

    }

    void turnOff(){

        System.out.println("this light is off");

    }

}

71.Abstraction 10th program:

class Abst10{

    //Create an abstract class Question with method displayQuestion().

    //Extend it to MCQQuestion and TrueFalseQuestion.

    public static void main(String[] args){

        MCQQuestion m = new MCQQuestion();

        TrueFalseQuestion t = new TrueFalseQuestion();

        m.displayQuestion();

        t.displayQuestion();

    }

}

abstract class Question{

    abstract void displayQuestion();

}

class MCQQuestion extends Question{

    void displayQuestion(){

        System.out.println("this is mcq ");

    }

}

class TrueFalseQuestion extends Question{

    void displayQuestion(){

        System.out.println("this is true and false");

    }

}

72.Abstraction 11th program:

class Abst11{

    //Create an abstract class Logger with an abstract method log(String message).

    //Implement FileLogger and ConsoleLogger.

    public static void main(String[] args){

        FileLogger f = new FileLogger();

        ConsoleLogger c = new ConsoleLogger();

        f.log();

        c.log();

    }

}

interface Logger{

    abstract void log();

}

class FileLogger implements Logger{

    public void log(){

        System.out.println("this is a String message");

    }

}

class ConsoleLogger implements Logger{

    public void log(){

        System.out.println("This is another String message");

    }

}

73.Abstraction 12th program:

class Abst12{

    //Create an abstract class MediaPlayer with method play() and stop().

    //Implement AudioPlayer and  VideoPlayer.

    public static void main(String[] args){

        AudioPlayer a = new AudioPlayer();

        VideoPlayer v = new VideoPlayer();

        a.play();

        a.stop();

        v.play();

        v.stop();

    }

}

interface MediaPlayer{

    abstract void play();

    abstract void stop();

}

class AudioPlayer implements MediaPlayer{

    public void play(){

        System.out.println("this is play of Audioplayer");

    }

    public void stop(){

        System.out.println("this is stop of audioplayer");

    }

}

class VideoPlayer implements MediaPlayer{

    public void play(){

        System.out.println("this is play of videopalyer");

    }

    public void stop(){

        System.out.println("This is stop of videoplayer");

    }

}

74.Exception handling 1st program:

class excep1{

    //Create a program that attempts to call a method on a null object

    //and handle the NullPointerException gracefully.

    public static void main(String[] args){

        excep1 x = new excep1();

        try{

            x.len();

        }catch(NullPointerException e){

            System.out.println("this is null. Their is no length exist");

        }

    }

    public void len(){

        String str = null;

        System.out.println("This is the length of str" + str.length());

    }

}

75.Exception handling 2nd program:

class excep2 {

    public static void main(String[] args){

        int a= 10;

        System.out.println("this is a "+ a);

        int b = 0;

        System.out.println("this is b " +b );

        try{

        int c = a/b;

        System.out.println("this is c " + c);

        }catch(ArithmeticException e){

        }

        finally{

            System.out.println("this block is always execute");

        }

    }

}

76.Exception handling 3rd program:

import java.util.Scanner;

class excep3{

    public static void main(String[] args){

        System.out.println("enter the string ");

        Scanner sc = new Scanner(System.in);

        String str = sc.nextLine();

        try{

        int number = Integer.parseInt(str);

        System.out.println("these are numbers " + number);

        }catch(Exception e){

            System.out.println("enter the valid numbers");

        }

    }

}

77.Exception handling 4th program:

class excep4{

    static void checkNum(int Number){

            if(Number <0){

                throw new ArithmeticException("this is not a positive no.");

            }

            else{

                System.out.println("This is a valid no." + Number);

            }

        }

    public static void main(String[]args){

        checkNum(5);

    }

}