## RAJALAKSHMI ENGINEERING COLLEGE

.

**[AUTONOMOUS]**

**RAJALAKSHMI NAGAR, THANDALAM – 602 105**





Name : .G.K.Sritharanika……..

Year / Branch / Section : . . II/IT/D.. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

College Roll No. : . . . . . . . 231001215 . . . . . . . . . . . . . . . . . . . . . . . . . .

Semester : . . . . . III. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

Academic Year : . . . . . . . . . . . 2024-2025 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .



**CS23333 OBJECT ORIENTED PROGRAMING USING JAVA**

**Laboratory Record Note Book**

## RAJALAKSHMI ENGINEERING COLLEGE

**[AUTONOMOUS]**

### RAJALAKSHMI NAGAR, THANDALAM – 602 105

**BONAFIDE CERTIFICATE**

Name : . .G.K.Sritharanika.. . . . . . . . . . . . . . . . . . . . . . .

Academic Year : 2024-2025 Semester:. III Branch : IT-D

**Register No.**

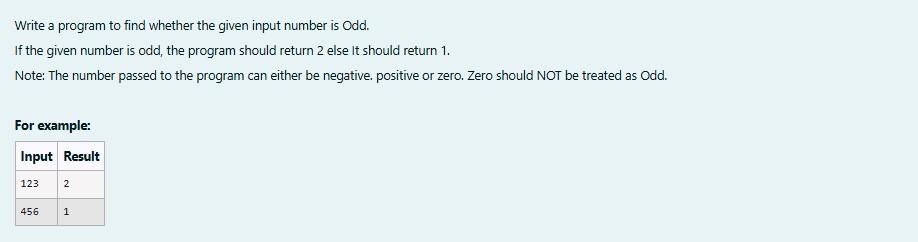
2116231001215

Certified that this is the bonafide record of work done by the above student in the CS23333 –Object Oriented Programming using JAVA during the year 2024 - 2025.

**Signature of Faculty in-charge**

Submitted for the Practical Examination held on . 27.11.2024. . . . . . . . . . . .

**Internal Examiner External Examiner**

1.

**SOLUTION :**

import java.util.Scanner; public class oddorEven{ public static void

main(String[]args){ Scanner s=new Scanner(System.in); int number = s.nextInt(); if(number %2==0){

System.out.println(1);

} else

{

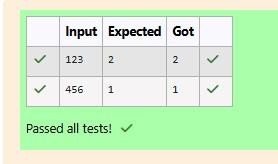
System.out.println(2);

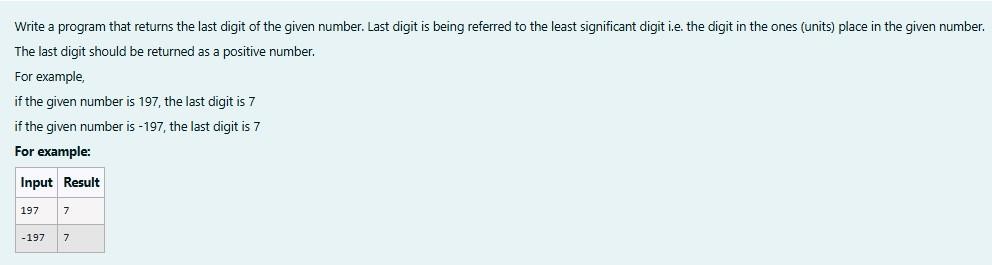
}

}

}

**OUTPUT :**



**2.**

**SOLUTION :**

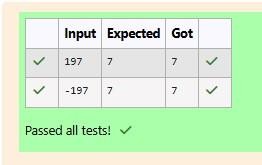
import java.util.Scanner; import java.lang.Math; public class LastDigit{ public static void main(String[]args){ Scanner s=new Scanner(System.in);

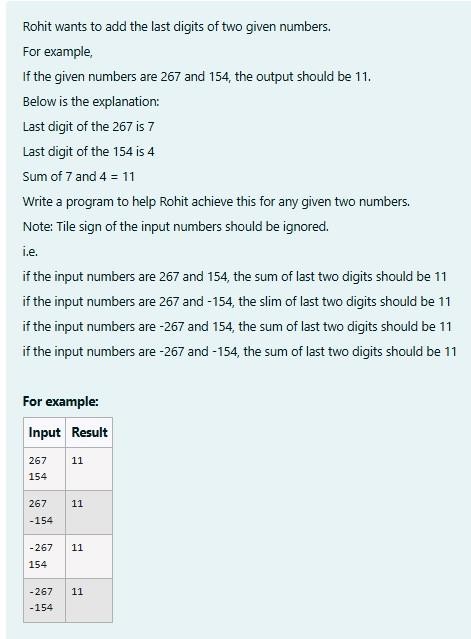
int a = s.nextInt(); int lastDigit=Math.abs(a%10); System.out.println(lastDigit);

}

}

**OUTPUT :**



**3.**

**SOLUTION :**

import java.util.Scanner; import java.lang.Math;

public class number{ public static void main(String[]args){ Scanner s= new Scanner(System.in);

int a = s.nextInt(); int b = s.nextInt();

System.out.println(Math.abs(a)%10+Math.abs(b)%10);

}

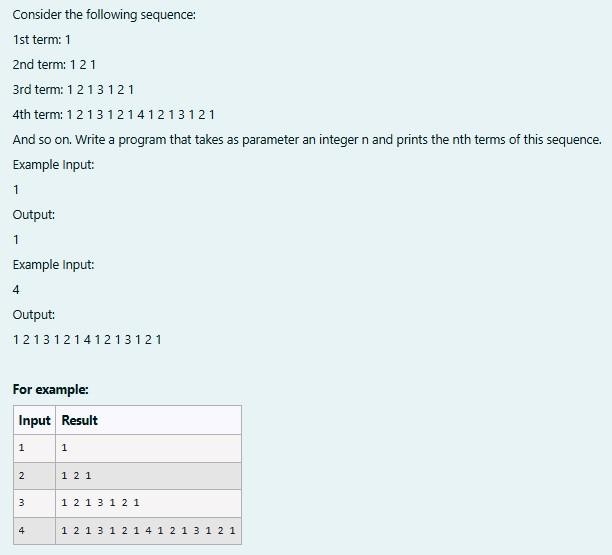
}

**OUTPUT:**



### [Lab-02-Flow](http://www.rajalakshmicolleges.org/moodle/course/section.php?id=50) [Control](http://www.rajalakshmicolleges.org/moodle/course/section.php?id=50) [Statements](http://www.rajalakshmicolleges.org/moodle/course/section.php?id=50)

**1.**



**SOLUTION :**

import java.util.Scanner; public class SequenceGenerator{ public static void main(String[]args){ Scanner S = new Scanner(System.in);

int n = S.nextInt();

String term = generateTerm(n); System.out.print(term);

}

private static String generateTerm(int n){ if (n==1){ return "1";

}

String prevTerm = generateTerm (n-1);

StringBuilder currentTerm = new StringBuilder(prevTerm);

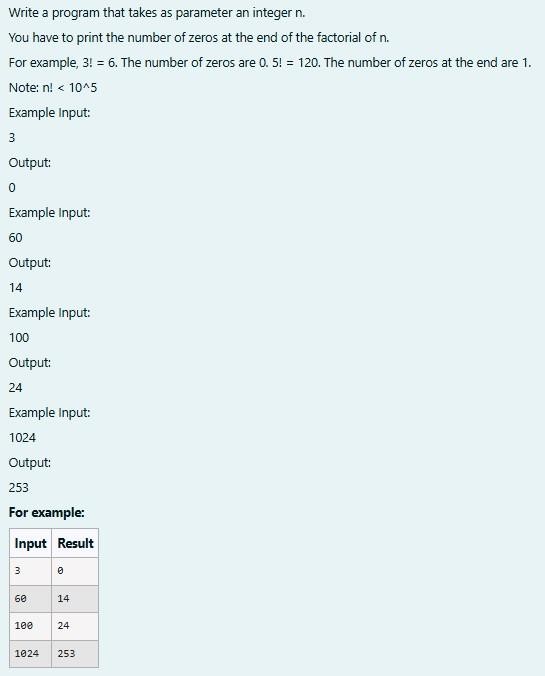
currentTerm.append(" " + n + " "); currentTerm.append(prevTerm); return currentTerm.toString();

}

}

**OUTPUT :**



**2.**

**SOLUTION :**

// Java program to count trailing 0s in n! import java.io.\*; import java.util.Scanner;

class prog {

// Function to return trailing

// 0s in factorial of n

static int findTrailingZeros(int n)

{ if (n < 0) // Negative Number Edge Case return -1;

// Initialize result

int count=0;

// Keep dividing n by powers // of 5 and update count for (int i = 5; n / i >= 1; i\*=5 ){ count

+= n / i;

} return count;

}

// Driver Code

public static void main(String[] args)

{

Scanner sc= new Scanner(System.in); int n=sc.nextInt();

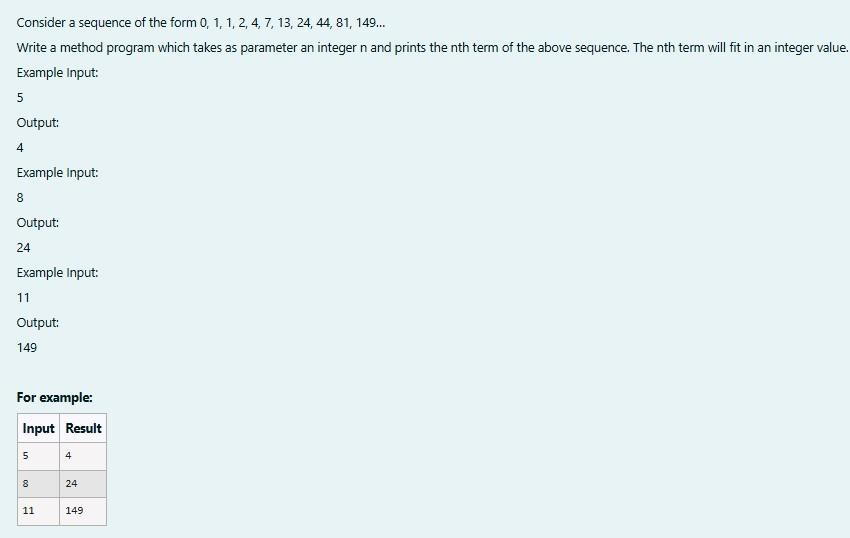
int res=findTrailingZeros(n); System.out.println(res);

}

}

**OUTPUT :**



**3.**

**SOLUTION :**

import java.util.Scanner; class fibo3{ int a; int b; int c;

fibo3(int a,int b,int c){ this.a = a; this.b = b; this.c = c;

}

int nth(int x){ if (x == 1){ return 0;

}

else if(x == 2 && x == 3) return 1;

else{ int temp1,temp2,temp; int count = 4; while(x >= count){ temp = this.a+this.b+this.c;

temp1 = this.c; this.c = temp; temp2 = this.b; this.b = temp1; this.a = temp2; count++;

}

return this.c;

}

}

}

public class Main{ public static void main(String[] args){ Scanner s = new Scanner(System.in);

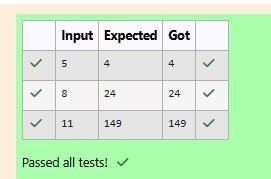
int t = s.nextInt(); fibo3 r

= new fibo3(0,1,1); System.out.print(r.nth(t));

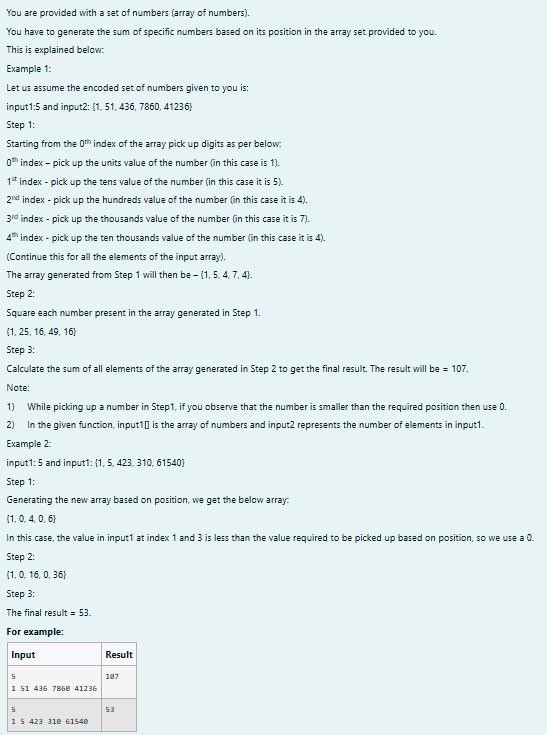
}

}

**OUTPUT :**



### [Lab-03-Arrays](http://www.rajalakshmicolleges.org/moodle/course/section.php?id=51)

**1.**

**SOLUTION :**

import java.util.Scanner; public class digit{ public static void main(String[]args){

Scanner scanner =new Scanner(System.in);

int size =scanner.nextInt(); int[]inpar=new int[size]; for(int i=0;i<size;i++){ inpar[i]=scanner.nextInt();

}

int[]dig=new int[size]; for(int i=0;i<size;i++){ int num=inpar[i]; if(i==0){ dig[i]=num%10;

}

else if (i==1){ dig[i]=(num/10)%10;

}

else if(i==2){ dig[i]=(num/100)%10;

}

else if(i==3){ dig[i]=(num/1000)%10;

}

else if(i==4){ dig[i]=(num/10000)%10;

} else{ dig[i]=0;

}

} int fin=0; for(int digi:dig){ fin+=digi\*digi;

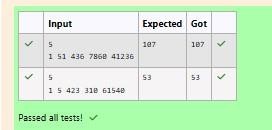
}

System.out.print(fin);

}

}

**OUTPUT :**



**2.**



**SOLUTION :**

}

import java.util.Scanner; public class longdig{ public static void main(String[]args){ Scanner sc=new Scanner(System.in);

int n=sc.nextInt();

int c = 1,v,seqtemp = 0,seq = 0,countmax = 0; int count = 0; while(c <= n){ v = sc.nextInt(); if(v >= 0){ countmax= countmax + v;

seqtemp++;

}

else{

seqtemp = 0;

countmax = 0;

}

if(seqtemp > seq ){ seq = seqtemp; count = countmax;

}

else if (seq == seqtemp){ count = count + countmax;

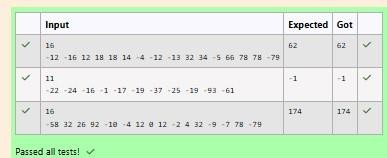
} c++; }

if (count == 0) System.out.print(-1);

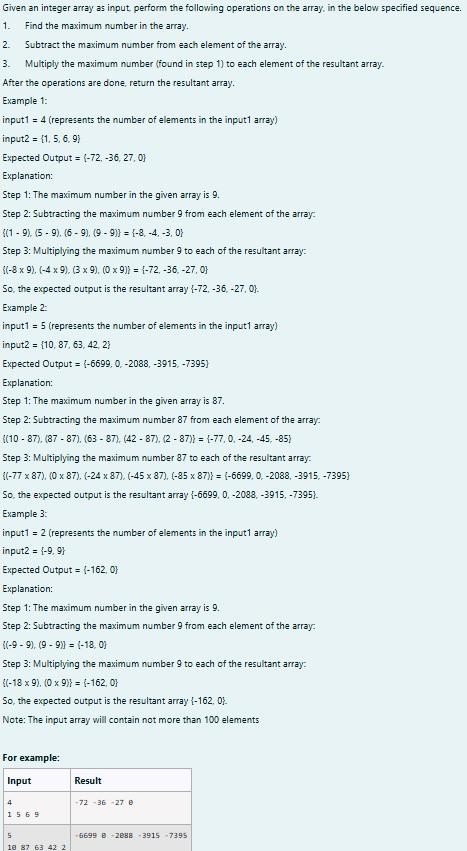
else

System.out.print(count);

}

**OUTPUT :**

**3.**



**SOLUTION :**

import java.util.Scanner; public class res{ public static int[]pa(int[]arr){

int maxs=Integer.MIN\_VALUE; for (int num:arr){

if(num>maxs){ maxs=num;

}

}

for(int i=0;i<arr.length;i++){ arr[i]=(arr[i]- maxs)\*maxs;

}

return arr;

}

public static void main(String[]args){

Scanner scanner =new Scanner (System.in); int n=scanner.nextInt();

int[]arr=new int[n]; for(int i=0;i<n;i++){ arr[i]=scanner.nextInt();

}

int[]res=pa(arr); for(int i=0;i<n;i++){

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

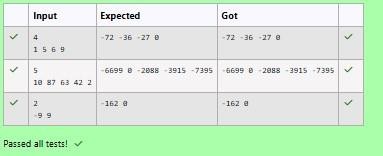
}

scanner.close();

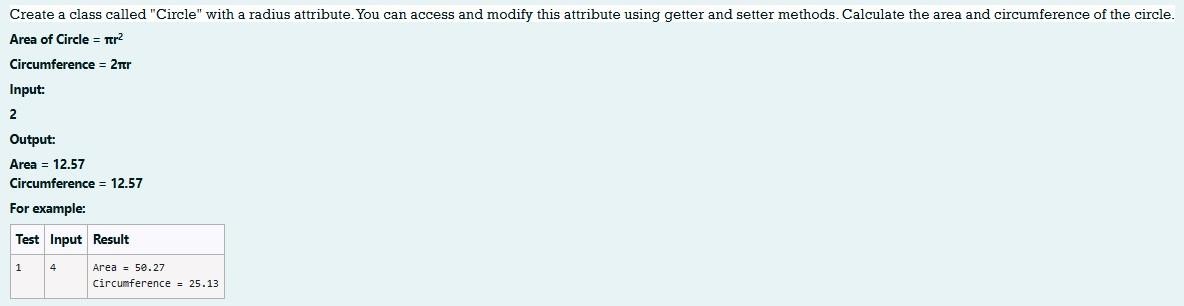
}

}

**OUTPUT :**



### [Lab-04-Classes](http://www.rajalakshmicolleges.org/moodle/course/section.php?id=52) [and](http://www.rajalakshmicolleges.org/moodle/course/section.php?id=52) [Objects](http://www.rajalakshmicolleges.org/moodle/course/section.php?id=52)

**1.**

**SOLUTION :**

import java.io.\*; import java.util.Scanner; class Circle

{ private double radius; public Circle(double radius){

// set the instance variable radius this.radius =radius;

} public void setRadius(double radius){

// set the radius this.radius=radius;

}

public double getRadius()

// return the radius

return radius;

{

}

public double calculateArea() { // complete the below statement return Math.PI\*radius\*radius;

}

public double calculateCircumference() {

// complete the statement return 2\*Math.PI\*radius;

}

} class prog{ public static void main(String[] args) { int r;

Scanner sc= new Scanner(System.in); r=sc.nextInt();

Circle c= new Circle(r);

System.out.println("Area = "+String.format("%.2f", c.calculateArea()));

// invoke the calculatecircumference method System.out.println("Circumference = "+String.format("%.2f" ,

c.calculateCircumference()));

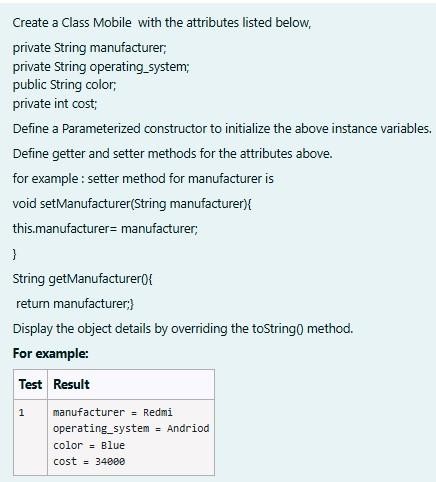
sc.close();

}

}

**OUTPUT :**



**2.**

**SOLUTION :**

public class mobile{ private String man; private String os; public String clr; private int cost;

public mobile(String man,String os,String clr,int cost){ this.man=man; this.os=os; this.clr=clr; this.cost=cost;

}

public String toString(){ return "manufacturer = "+man+"\n"+"operating\_system = "+os+"\n"+"color = "+ clr+"\n"+"cost = "+cost;

}

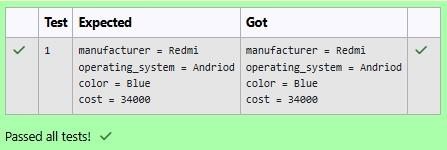
public static void main(String[]args){

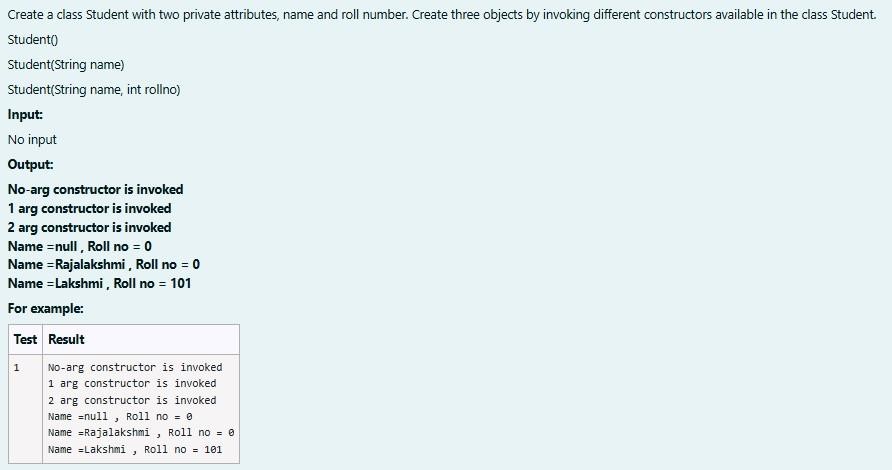
mobile mobile=new mobile("Redmi","Andriod","Blue",34000); System.out.println(mobile);

}

}

**OUTPUT :**



**3.**

**SOLUTION :**

public class stud{ private String name; private int roll; public stud(){

System.out.println("No-arg constructor is invoked"); name=null; roll=0;

}

public stud(String name){

System.out.println("1 arg constructor is invoked"); this.name=name; roll=0;

}

public stud(String name,int roll){

System.out.println("2 arg constructor is invoked"); this.name=name; this.roll=roll;

}

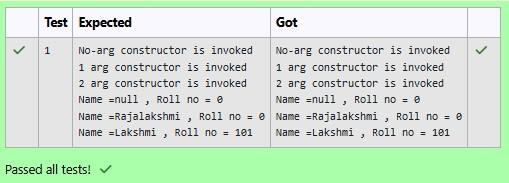
public static void main (String[]args){ stud s1=new stud(); stud s2=new stud("Rajalakshmi"); stud s3=new stud("Lakshmi",101);

System.out.println("Name ="+s1.name+" , Roll no = "+s2.roll); System.out.println("Name ="+s2.name+" , Roll no = "+s2.roll); System.out.println("Name ="+s3.name+" , Roll no = "+s3.roll);

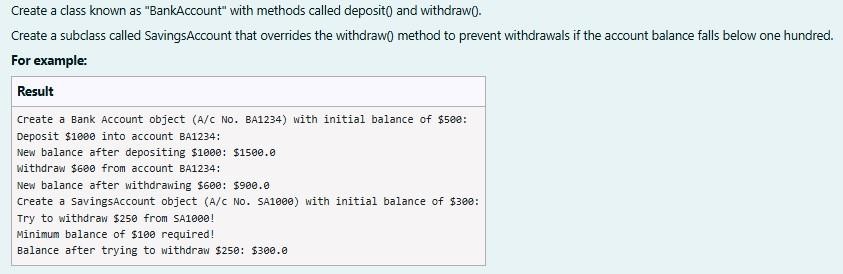
}

}

**OUTPUT :**



### [Lab-05-Inheritance](http://www.rajalakshmicolleges.org/moodle/course/section.php?id=55)

**1.**

**SOLUTION :**

class BankAccount {

// Private field to store the account number private String accountNumber;

// Private field to store the balance

private double balance;

// Constructor to initialize account number and balance public BankAccount(String accountNumber,double balance){ this.accountNumber=accountNumber;

this.balance=balance;

}

// Method to deposit an amount into the account public void deposit(double amount) {

// Increase the balance by the deposit amount balance+=amount;

}

// Method to withdraw an amount from the account public void withdraw(double amount) {

// Check if the balance is sufficient for the withdrawal if (balance >= amount) {

// Decrease the balance by the withdrawal amount balance -= amount;

} else {

// Print a message if the balance is insufficient System.out.println("Insufficient balance"); }

}

// Method to get the current balance public double getBalance() { // Return the current balance

return balance;

}

public String getAccountNumber(){ return accountNumber;

}

}

class SavingsAccount extends BankAccount {

// Constructor to initialize account number and balance

public SavingsAccount(String accountNumber, double balance) {

// Call the parent class constructor super(accountNumber,balance);

}

// Override the withdraw method from the parent class @Override

public void withdraw(double amount) {

// Check if the withdrawal would cause the balance to drop below $100

if (getBalance() - amount < 100) {

// Print a message if the minimum balance requirement is not met System.out.println("Minimum balance of $100 required!");

} else {

// Call the parent class withdraw method super.withdraw(amount);

}

}

} public class Main {

public static void main(String[] args) {

// Print message to indicate creation of a BankAccount object System.out.println("Create a Bank Account object (A/c No. BA1234) with initial

balance of $500:");

// Create a BankAccount object (A/c No. "BA1234") with initial balance of $500 BankAccount BA1234 = new BankAccount("BA1234", 500);

// Print message to indicate deposit action System.out.println("Deposit $1000 into account BA1234:");

// Deposit $1000 into account BA1234 BA1234.deposit(1000);

// Print the new balance after deposit

System.out.println("New balance after depositing $1000: $"+BA1234.getBalance());

// Print message to indicate withdrawal action System.out.println("Withdraw $600 from account BA1234:");

// Withdraw $600 from account BA1234 BA1234.withdraw(600);

// Print the new balance after withdrawal System.out.println("New balance after withdrawing $600: $" +

BA1234.getBalance());

// Print message to indicate creation of another SavingsAccount object System.out.println("Create a SavingsAccount object (A/c No. SA1000) with initial

balance of $300:");

// Create a SavingsAccount object (A/c No. "SA1000") with initial balance of $300 SavingsAccount SA1000 = new SavingsAccount("SA1000", 300);

// Print message to indicate withdrawal action System.out.println("Try to withdraw $250 from SA1000!");

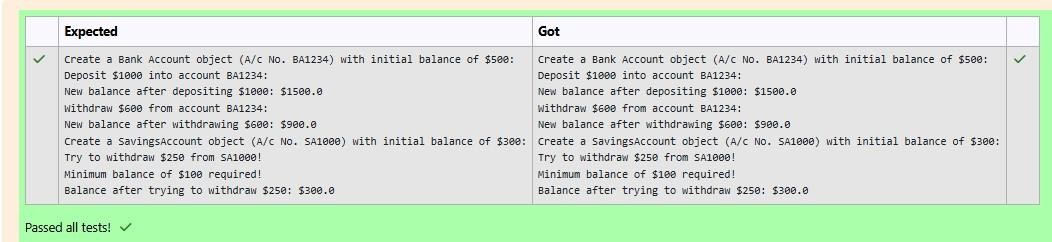
// Withdraw $250 from SA1000 (balance falls below $100) SA1000.withdraw(250);

// Print the balance after attempting to withdraw $250 System.out.println("Balance after trying to withdraw $250: $" +

SA1000.getBalance()); }

}

**OUTPUT :**



**2.**

**SOLUTION :**

class College

{

public String collegeName;

public College(String collegeName)

{ // initialize the instance variables this.collegeName=collegeName; }

public void admitted() {

System.out.println("A student admitted in "+collegeName);

} } class Student extends College{

String studentName;

String department;

public Student(String collegeName, String studentName,String department) {

// initialize the instance variables super(collegeName); this.studentName=studentName; this.department=department;

}

public String toString(){

// return the details of the student return "CollegeName : "+collegeName+"\n"+"StudentName : "+studentName+"\n"+"Department : "+department;

}

}

public class Main {

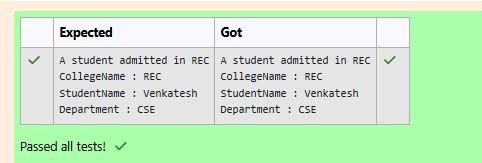
public static void main (String[] args) {

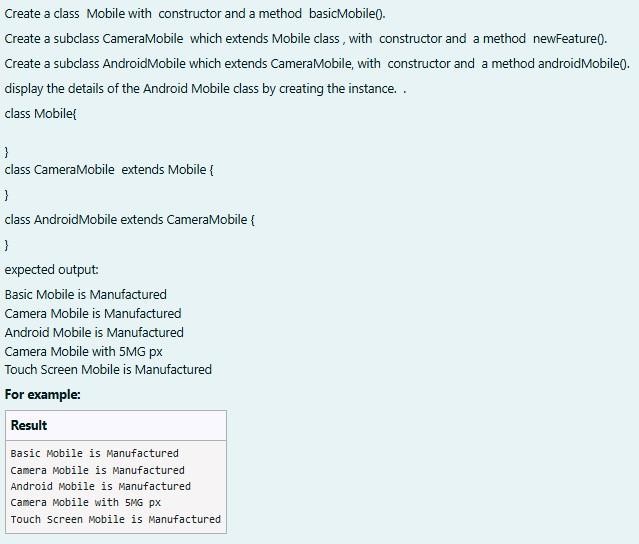
Student s1 = new Student("REC","Venkatesh","CSE"); s1.admitted(); // invoke the admitted() method System.out.println(s1.toString());

}

}

**OUTPUT :**



**3.**

**SOLUTION :**

class mob{ mob(){

System.out.println("Basic Mobile is Manufactured");

}

void basmob(){

System.out.println("Basic Mobile is Manufactured");

}

}

class cam extends mob{ cam(){

super();

System.out.println("Camera Mobile is Manufactured");

}

void newm(){

System.out.println("Camera Mobile with 5MG px");

}

}

class and extends cam{ and(){

super();

System.out.println("Android Mobile is Manufactured");

}

void andmob(){

System.out.println("Touch Screen Mobile is Manufactured");

}

} public class Main{ public static void main(String[]args){ and andmob=new and(); andmob.newm(); andmob.andmob();

}

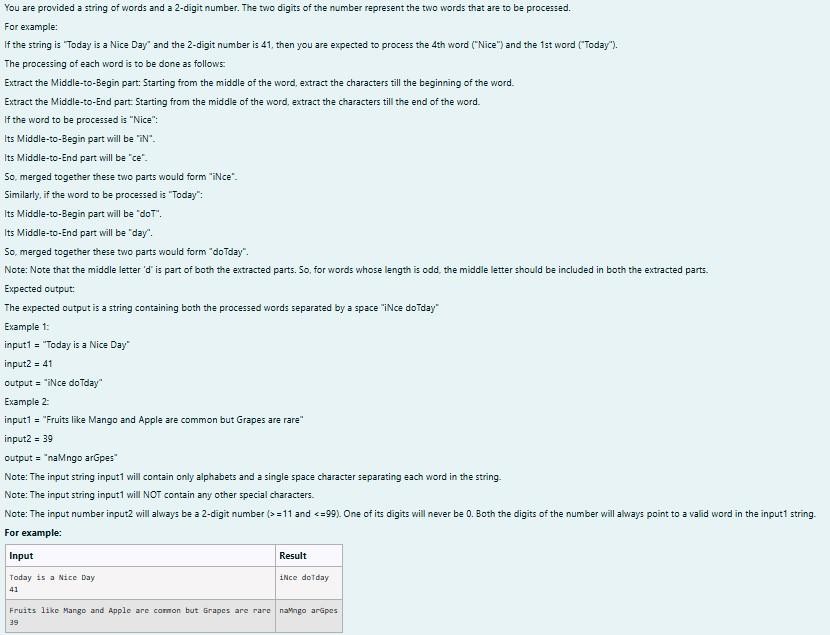
}

**OUTPUT :**



### [Lab-06-String,](http://www.rajalakshmicolleges.org/moodle/course/section.php?id=54) [StringBuffer](http://www.rajalakshmicolleges.org/moodle/course/section.php?id=54)

**1.**



**SOLUTION :**

import java.util.\*; public class mix{ public static void main(String[] args){

Scanner scan = new Scanner(System.in); String g = scan.nextLine(); int n = scan.nextInt(),ones,flag = 0; StringBuffer temp = new StringBuffer(); StringBuffer temp1 = new StringBuffer(); int space = 0; while (n > 0){ ones = (n %10) - 1;

for(int i = 0; i < g.length();i++){ if (g.charAt(i) == ' '){ space

= space + 1;

}

else if(space == ones && flag == 0){ temp.append(Character.toString(g.charAt(i)));

}

else if(space == ones && flag == 1){ temp1.append(Character.toString(g.charAt(i)));

}

} space = 0 ; flag = 1; n = n

/10;

}

rew m = new rew();

System.out.println(m.r(temp1.toString()) + " " + m.r(temp.toString()));

}

}

class rew{

String r(String a){ int le

= a.length(),n,q;

StringBuffer temp3 = new StringBuffer(); if(le % 2 == 1){

n = ((int)(le/2));

q = ((int)(le/2));

} else{ n = ((int)(le/2)) - 1;

q = ((int)(le/2));

}

for(int i = n;i >= 0;i--){ temp3.append(Character.toString(a.charAt(i)));

} for(int i = q;i < le;i++){ temp3.append(Character.toString(a.charAt(i)));

}

return temp3.toString();

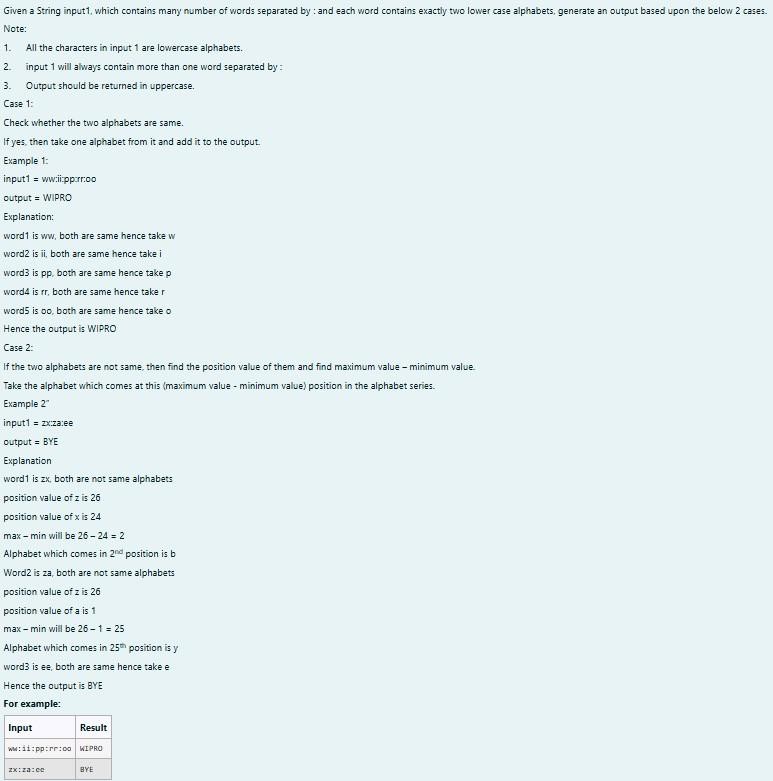
}

}

**OUTPUT :**



**2.**



**SOLUTION :**

import java.util.\*; class diff{ char different(char a, char b){ if ((int)a != (int)b) return (char)((int)'a' + ((int)a-(int)b) - 1);

return a;

}

}

public class Main{ public static void main(String[] args){ Scanner scan = new Scanner(System.in);

diff z = new diff();

String q = scan.nextLine(); StringBuffer ans = new StringBuffer(); StringBuffer temp = new StringBuffer(); for(int i = 0;i < q.length();i++){ if(q.charAt(i) == ':'){ temp.append(" ");

} else{ temp.append(Character.toString(q.charAt(i))); }

}

String h = temp.toString(); for(int i

= 0;i < temp.length();i++){ if(i%3

== 0){

ans.append(Character.toString(z.different(h.charAt(i),h.charAt(i+1))));

}

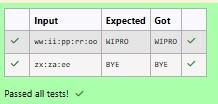
}

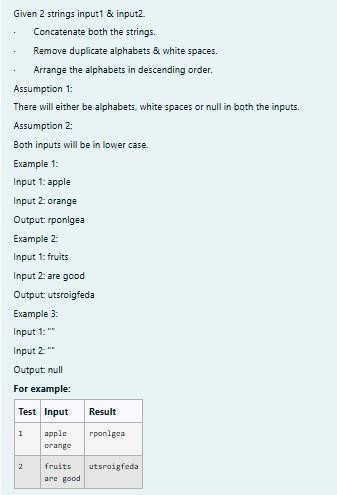
System.out.print(ans.toString().toUpperCase());

}

}

**OUTPUT :**



**3.**

**SOLUTION :**

import java.util.\*;

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

Scanner scan = new Scanner(System.in); String a = scan.nextLine();

String b = scan.nextLine(); StringBuffer ab = new StringBuffer();

if(a.trim().isEmpty() && b.trim().isEmpty()){ System.out.print("null");

}

else{

for(int i = 0;i < a.length();i++){ if (a.charAt(i)

!= ' ') {

ab.append(Character.toString(a.charAt(i))); }

}

for(int i = 0;i < b.length();i++){ if (b.charAt(i)

!= ' '){

ab.append(Character.toString(b.charAt(i))); }

}

char[] d = ab.toString().toCharArray(); Arrays.sort(d);

for(int i = d.length - 1;i >= 1;i--){ if(d[i] != d[i-1])

System.out.print(d[i]);

}

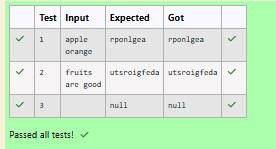
System.out.print(d[0]);

}

}

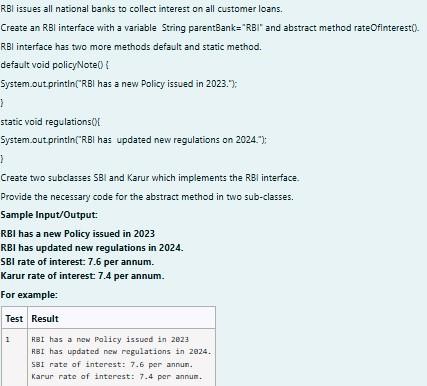
}

**OUTPUT :**



### [Lab-07-Interfaces](http://www.rajalakshmicolleges.org/moodle/course/section.php?id=58)

**1.**



**SOLUTION :**

// Define the RBI interface interface RBI {

// Variable declaration String parentBank = "RBI";

// Abstract method double rateOfInterest();

// Default method

default void policyNote() {

System.out.println("RBI has a new Policy issued in 2023");

}

// Static method

static void regulations() {

System.out.println("RBI has updated new regulations in 2024.");

}

}

// SBI class implementing RBI interface class SBI implements RBI {

// Implementing the abstract method public double rateOfInterest() {

return 7.6;

}

}

// Karur class implementing RBI interface class Karur implements RBI { // Implementing the abstract method public double rateOfInterest() { return 7.4;

}

}

// Main class to test the functionality public class Main { public static void main(String[] args) {

// RBI policies and regulations

RBI rbi = new SBI(); // Can be any class implementing RBI rbi.policyNote(); // Default method RBI.regulations();

// Static method

// SBI bank details SBI sbi = new SBI();

System.out.println("SBI rate of interest: " + sbi.rateOfInterest() + " per annum.");

// Karur bank details

Karur karur = new Karur();

System.out.println("Karur rate of interest: " + karur.rateOfInterest() + " per annum.");

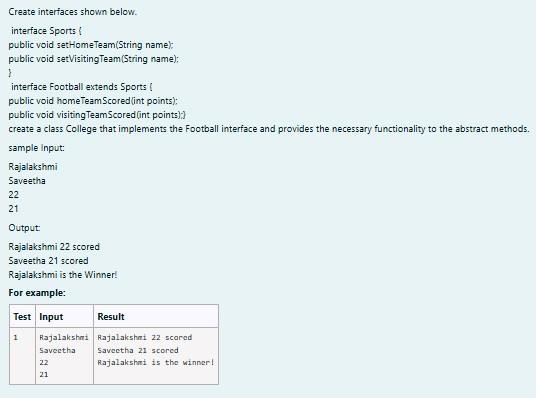
}

}

**OUTPUT :**



**2.**



**SOLUTION :**

import java.util.Scanner;

interface Sports { void setHomeTeam(String name); void setVisitingTeam(String name);

}

interface Football extends Sports { void homeTeamScored(int points); void visitingTeamScored(int points);

}

class College implements Football { private String homeTeam; private String visitingTeam; private int homeTeamPoints = 0; private int visitingTeamPoints = 0;

public void setHomeTeam(String name) { this.homeTeam = name;

}

public void setVisitingTeam(String name) { this.visitingTeam = name;

} public void homeTeamScored(int points)

{

homeTeamPoints += points; System.out.println(homeTeam + " " + points + " scored");

}

public void visitingTeamScored(int points) { visitingTeamPoints += points; System.out.println(visitingTeam + " " + points + " scored");

}

public void winningTeam() { if (homeTeamPoints > visitingTeamPoints) {

System.out.println(homeTeam + " is the winner!");

} else if (homeTeamPoints < visitingTeamPoints) { System.out.println(visitingTeam + " is the winner!");

} else {

System.out.println("It's a tie match.");

}

}

}

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

Scanner sc = new Scanner(System.in);

// Get home team name String hname = sc.nextLine();

// Get visiting team name String vteam = sc.nextLine();

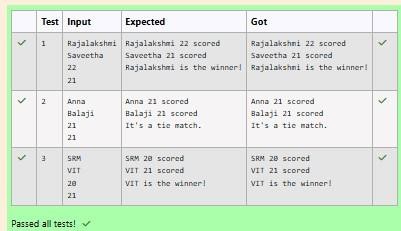
// Create College object College match = new College(); match.setHomeTeam(hname); match.setVisitingTeam(vteam);

// Get points scored by home team int htpoints = sc.nextInt(); match.homeTeamScored(htpoints);

// Get points scored by visiting team int vtpoints = sc.nextInt(); match.visitingTeamScored(vtpoints);

// Determine and print the winning team match.winningTeam();

sc.close();

**OUTPUT :**

**3.**



**SOLUTION :**

import java.util.Scanner;

// Define the Playable interface interface Playable {

// Abstract method to play the respective sport void play();

}

// Football class implementing Playable interface class Football implements Playable {

String name;

// Constructor

public Football(String name) { this.name = name;

}

// Override the play method

public void play() {

System.out.println(name + " is Playing football");

}

}

// Volleyball class implementing Playable interface class Volleyball implements Playable {

String name;

// Constructor

public Volleyball(String name) { this.name = name;

}

// Override the play method public void play() {

System.out.println(name + " is Playing volleyball");

}

}

// Basketball class implementing Playable interface class Basketball implements Playable {

String name;

// Constructor

public Basketball(String name) { this.name = name;

}

// Override the play method public void play() {

System.out.println(name + " is Playing basketball");

}

}

// Main class to test the functionality public class Main { public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

// Input for Football player

String footballPlayerName = scanner.nextLine();

Football footballPlayer = new Football(footballPlayerName);

// Input for Volleyball player

String volleyballPlayerName = scanner.nextLine();

Volleyball volleyballPlayer = new Volleyball(volleyballPlayerName);

// Input for Basketball player

String basketballPlayerName = scanner.nextLine();

Basketball basketballPlayer = new Basketball(basketballPlayerName);

// Call the play method for each player footballPlayer.play(); volleyballPlayer.play(); basketballPlayer.play();

scanner.close();

}

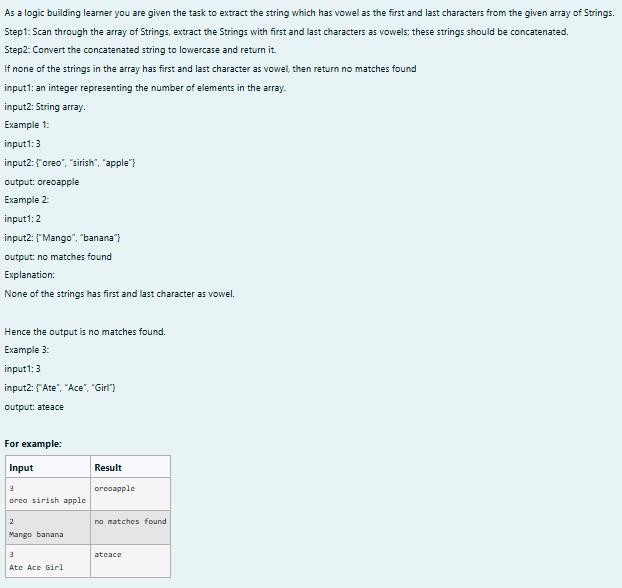
}

**OUTPUT :**



### [Lab-08](http://www.rajalakshmicolleges.org/moodle/course/section.php?id=57) [-](http://www.rajalakshmicolleges.org/moodle/course/section.php?id=57) [Polymorphism,](http://www.rajalakshmicolleges.org/moodle/course/section.php?id=57) [Abstract](http://www.rajalakshmicolleges.org/moodle/course/section.php?id=57) [Classes,](http://www.rajalakshmicolleges.org/moodle/course/section.php?id=57) [final](http://www.rajalakshmicolleges.org/moodle/course/section.php?id=57) [Keyword](http://www.rajalakshmicolleges.org/moodle/course/section.php?id=57)

**1.**



**SOLUTION :**

import java.util.Scanner; public class VowelStringExtractor {

// Method to extract strings with vowels as first and last characters public static String extractVowelStrings(String[] stringArray) {

StringBuilder result = new StringBuilder();

String vowels = "aeiouAEIOU"; // String containing all vowels

// Iterate through the array of strings for (String s : stringArray) {

// Check if the string is not empty and if both the first and last characters are vowels if (s.length() > 0 && vowels.indexOf(s.charAt(0)) != -1 && vowels.indexOf(s.charAt(s.length() - 1)) != -1) { result.append(s); // Append matching string to the result }

}

// Return the concatenated string in lowercase or "no matches found"

return result.length() > 0 ? result.toString().toLowerCase() : "no matches found"; }

public static void main(String[] args) { Scanner scanner = new Scanner(System.in);

// Input for the number of strings int n = scanner.nextInt();

scanner.nextLine(); // Consume the newline character

// Input for the strings in one line String input = scanner.nextLine();

String[] strings = input.split(" "); // Split input into an array

// Process and output the result

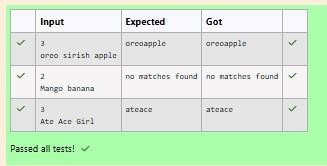
String result = extractVowelStrings(strings); System.out.println(result);

scanner.close(); // Close the scanner

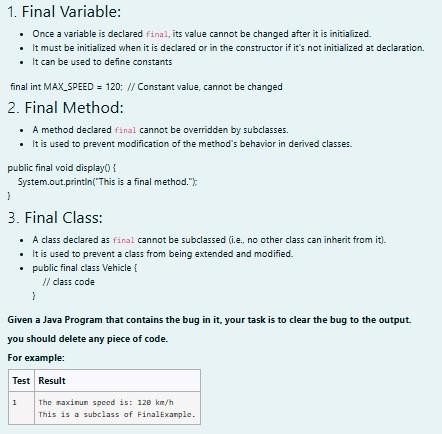
}

}

**OUTPUT :**



**2.**



**SOLUTION :**

// Final class definition final class FinalExample {

// Final variable

final int MAX\_SPEED = 120; // Constant value

// Final method

public final void display() {

System.out.println("The maximum speed is: " + MAX\_SPEED + " km/h");

}

}

// Main class to test the final class public class Test { public static void main(String[] args) {

// Create an instance of FinalExample FinalExample example = new FinalExample(); example.display();

// Uncommenting the following line will result in a compile-time error

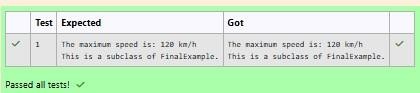
// because FinalExample is a final class and cannot be subclassed. // class SubclassExample extends FinalExample { }

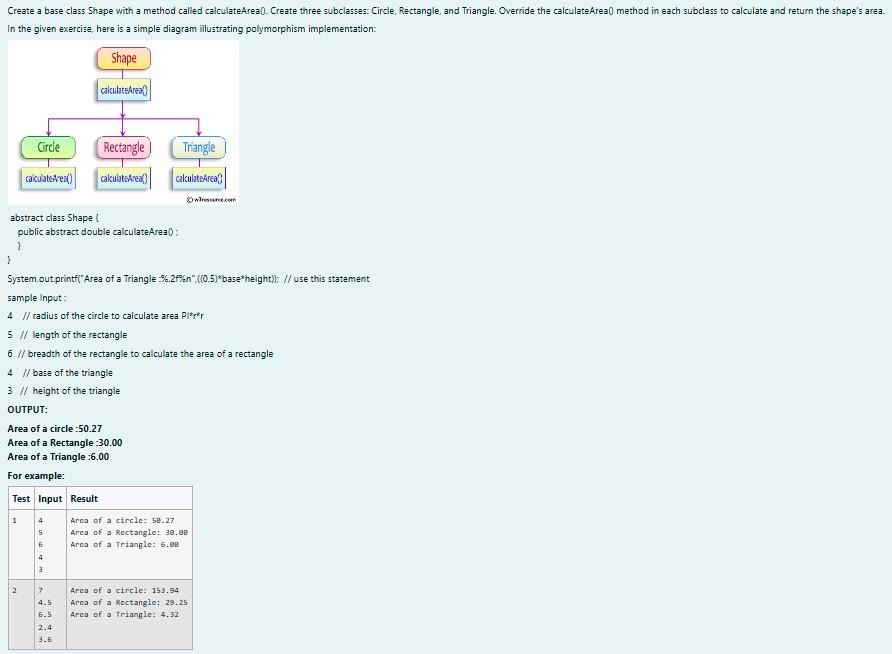
System.out.println("This is a subclass of FinalExample.");

}

}

**OUTPUT :**



**3.**

**SOLUTION :**

import java.util.Scanner;

// Abstract class Shape abstract class Shape { public abstract double calculateArea();

}

// Circle class

class Circle extends Shape { private double radius;

public Circle(double radius) { this.radius = radius;

}

@Override

public double calculateArea() { return Math.PI \* radius

\* radius; // Area of circle: πr² }

}

// Rectangle class

class Rectangle extends Shape { private double length; private double breadth;

public Rectangle(double length, double breadth) { this.length = length; this.breadth = breadth;

}

@Override

public double calculateArea() { return length \* breadth; // Area of rectangle: length \* breadth

}

}

// Triangle class

class Triangle extends Shape { private double base; private double height;

public Triangle(double base, double height) { this.base = base; this.height = height;

}

@Override

public double calculateArea() { return 0.5 \* base \* height; // Area of triangle: 0.5 \* base \* height

}

}

// Main class to test the shapes public class ShapeTest { public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

// Input for Circle

double radius = scanner.nextDouble(); Circle circle = new Circle(radius);

System.out.printf("Area of a circle: %.2f%n", circle.calculateArea());

// Input for Rectangle

double length = scanner.nextDouble(); double breadth = scanner.nextDouble();

Rectangle rectangle = new Rectangle(length, breadth); System.out.printf("Area of a Rectangle: %.2f%n", rectangle.calculateArea());

// Input for Triangle double base = scanner.nextDouble();

double height = scanner.nextDouble(); Triangle triangle = new Triangle(base, height);

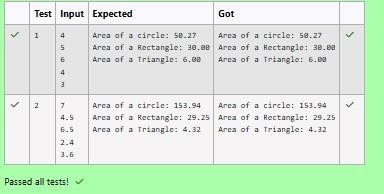
System.out.printf("Area of a Triangle: %.2f%n", triangle.calculateArea());

scanner.close();

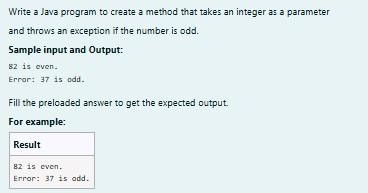
}

}

**OUTPUT :**



### [Lab-09-Exception](http://www.rajalakshmicolleges.org/moodle/course/section.php?id=59) [Handling](http://www.rajalakshmicolleges.org/moodle/course/section.php?id=59)

**1.**

**SOLUTION :**

class prog {

public static void main(String[] args) {

int n = 82; trynumber(n); n = 37;

trynumber(n); // Call the trynumber(n);

}

public static void trynumber(int n) { try { checkEvenNumber(n); // Call the checkEvenNumber() System.out.println(n + " is even.");

} catch (Exception e) { // Catch the exception System.out.println("Error: " + e.getMessage());

}

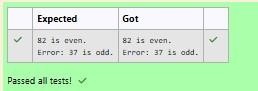
}

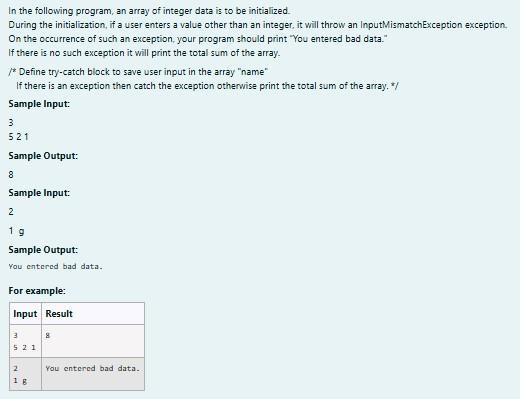
public static void checkEvenNumber(int number) { if (number % 2 != 0) { throw new RuntimeException(number + " is odd."); // Throw a RuntimeException }

**}**

**}**

**OUTPUT :**



**2.**

import java.util.Scanner;

import java.util.InputMismatchException;

class prog { public static void main(String[] args) { Scanner sc = new Scanner(System.in); int length = sc.nextInt();

// create an array to save user input int[] name = new int[length]; int sum = 0; // save the total sum of the array.

/\* Define try-catch block to save user input in the array "name" If there is an exception then catch the exception otherwise print the total sum of the array. \*/

try { for (int i = 0; i < length; i++) { name[i] = sc.nextInt(); // save user input in the array

}

// Calculate the total sum for (int num : name) {

sum += num;

}

// Print the total sum System.out.println(sum);

} catch (InputMismatchException e) { System.out.println("You entered bad data.");

}

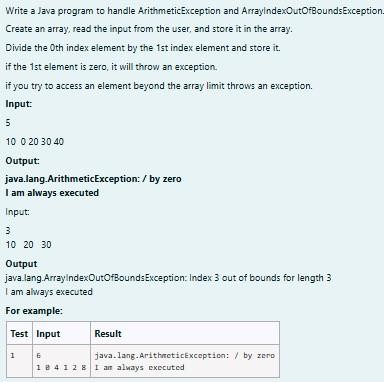
sc.close(); // Close the scanner

}

}

**OUTPUT :**





**SOLUTION :**

import java.util.Scanner;

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

Scanner scanner = new Scanner(System.in);

// Read the size of the array int size = scanner.nextInt();

// Initialize the array int[] numbers = new int[size];

// Read the elements into the array for (int i = 0; i < size; i++) { numbers[i] = scanner.nextInt();

}

try {

// Attempt to perform division

int result = numbers[0] / numbers[1]; // This may cause an ArithmeticException

} catch (ArithmeticException e) { System.out.println(e); // Catch division by zero

} catch (ArrayIndexOutOfBoundsException e) { System.out.println(e); // Catch accessing out of bounds

} catch (Exception e) {

System.out.println(e); // Catch any other exceptions

} finally {

// This block is always executed

}

try {

// Attempt to access an out-of-bounds index

int outOfBoundsValue = numbers[3]; // This will trigger ArrayIndexOutOfBoundsException if size < 4

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

} finally {

// This block is always executed for the second try System.out.println("I am always executed");

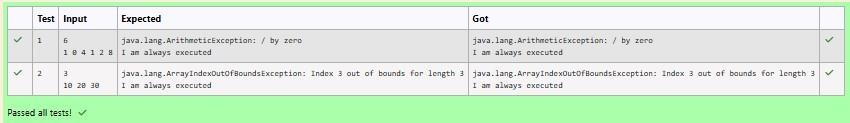
}

scanner.close();

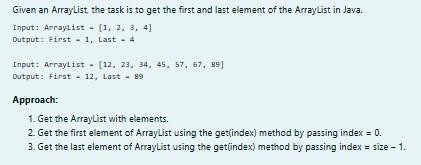
}

}

**OUTPUT :**



### [Lab-10-](http://www.rajalakshmicolleges.org/moodle/course/section.php?id=60) [Collection-](http://www.rajalakshmicolleges.org/moodle/course/section.php?id=60) [List](http://www.rajalakshmicolleges.org/moodle/course/section.php?id=60)

**1.**

**SOLUTION :**

import java.util.ArrayList; import java.util.Scanner;

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

Scanner scanner = new Scanner(System.in);

// Create an ArrayList

ArrayList<Integer> numbers = new ArrayList<>();

int numElements = scanner.nextInt();

for (int i = 0; i < numElements; i++) { int number = scanner.nextInt(); numbers.add(number);

}

System.out.println("ArrayList: " + numbers);

// Get the first element int firstElement = numbers.get(0);

// Get the last element

int lastElement = numbers.get(numbers.size() - 1);

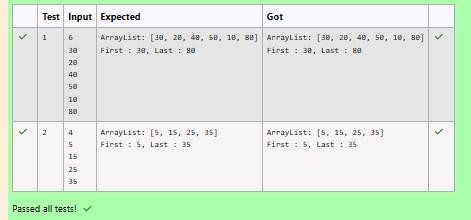
// Print the results

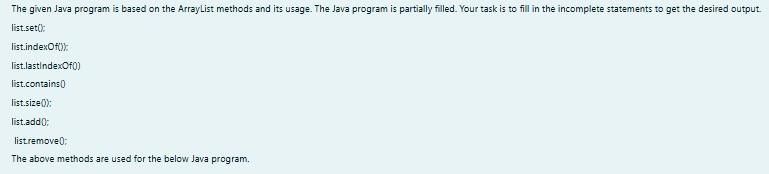
System.out.print("First : " + firstElement); System.out.println(", Last : " + lastElement);

}

}

**OUTPUT :**



**2.**

**SOLUTION :**

import java.util.ArrayList; import java.util.Scanner; public class Prog {

public static void main(String[] args)

{

Scanner sc= new Scanner(System.in); int n = sc.nextInt();

ArrayList<Integer> list = new ArrayList<Integer>(); for(int i = 0; i<n;i++)

list.add(sc.nextInt());

// printing initial value ArrayList System.out.println("ArrayList: " + list);

//Replacing the element at index 1 with 100 list.set(1,100);

//Getting the index of first occurrence of 100 System.out.println("Index of 100 = "+ list.indexOf(100)

);

//Getting the index of last occurrence of 100 System.out.println("LastIndex of 100 = "+ list.lastIndexOf(100));

// Check whether 200 is in the list or not System.out.println(list.contains(200)); //Output : false

// Print ArrayList size

System.out.println("Size Of ArrayList = "+list.size() );

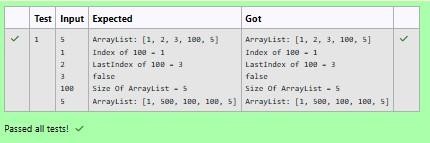
//Inserting 500 at index 1

list.add(1,500); // code here

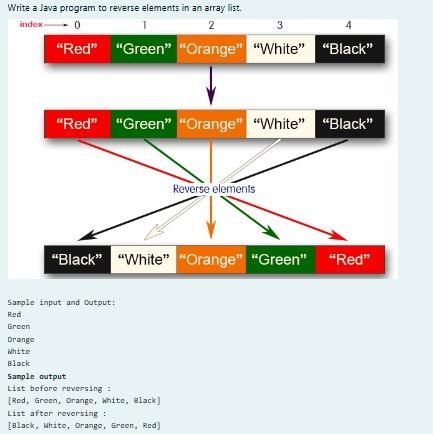
//Removing an element from position 3 list.remove(3); // code here System.out.print("ArrayList: " + list);

}

}

**OUTPUT :**

**3.**



**SOLUTION :**

import java.util.ArrayList; import java.util.Collections; import java.util.Scanner;

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

Scanner scanner = new Scanner(System.in);

ArrayList<String> list = new ArrayList<>(); int n = scanner.nextInt();

for (int i = 0; i < n; i++) {

String element = scanner.next(); list.add(element);

}

System.out.println("List before reversing : "); System.out.println(list);

Collections.reverse(list);

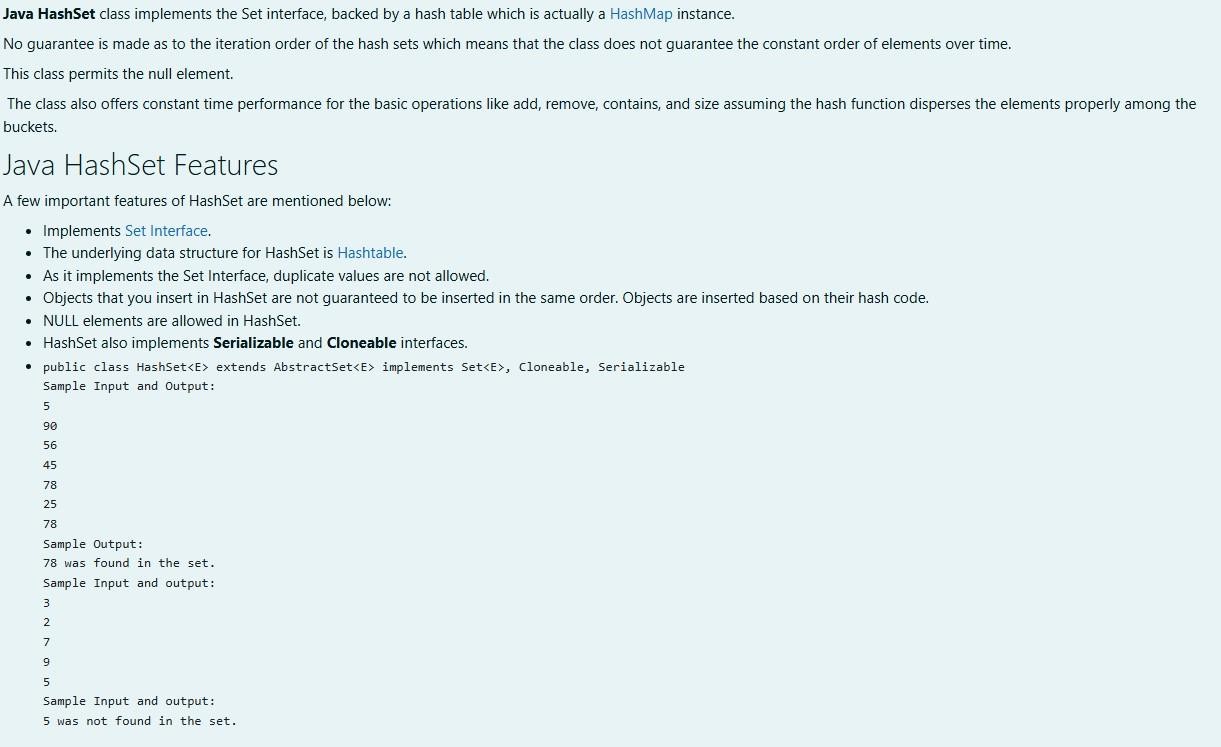
System.out.println("List after reversing : "); System.out.println(list);

}

}

**OUTPUT :**

### [Lab-11-Set,](http://www.rajalakshmicolleges.org/moodle/course/section.php?id=61) [Map](http://www.rajalakshmicolleges.org/moodle/course/section.php?id=61)

**1.**

**SOLUTION :**

**import java.util.HashSet; import java.util.Scanner;**

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

**Scanner sc = new Scanner(System.in);**

**// Read the number of elements int n = sc.nextInt();**

**// Create a HashSet object to store numbers HashSet<Integer> numbers = new HashSet<>();**

**// Add numbers to the HashSet for (int i = 0; i < n; i++) { numbers.add(sc.nextInt());**

**}**

**// Read the search key int skey = sc.nextInt();**

**// Check if skey is present in the HashSet if (numbers.contains(skey)) {**

**System.out.println(skey + " was found in the set.");**

**} else {**

**System.out.println(skey + " was not found in the set.");**

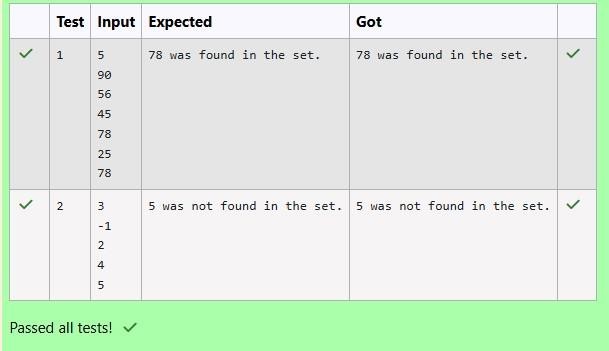
**}**

**// Close the scanner sc.close();**

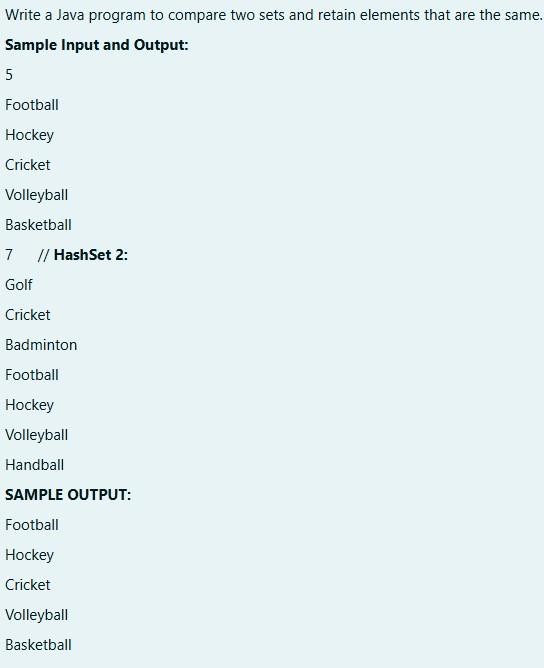
**}**

**}**

**OUTPUT :**



**2.**



**SOLUTION :**

**import java.util.HashSet; import java.util.Scanner; import java.util.Set;**

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

**Scanner scanner = new Scanner(System.in);**

**// Read the size of the first set**

**int size1 = Integer.parseInt(scanner.nextLine());**

**// Create a HashSet to store the first set of elements Set<String> set1 = new HashSet<>();**

**// Read elements for the first set for (int i = 0; i < size1; i++) { set1.add(scanner.nextLine());**

**}**

**// Read the size of the second set**

**int size2 = Integer.parseInt(scanner.nextLine());**

**// Create a HashSet to store the second set of elements Set<String> set2 = new HashSet<>();**

**// Read elements for the second set for (int i = 0; i < size2; i++) { set2.add(scanner.nextLine());**

**}**

**// Retain common elements using the retainAll() method set1.retainAll(set2);**

**// Print the common elements for (String element : set1) {**

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

**}**

**scanner.close();**

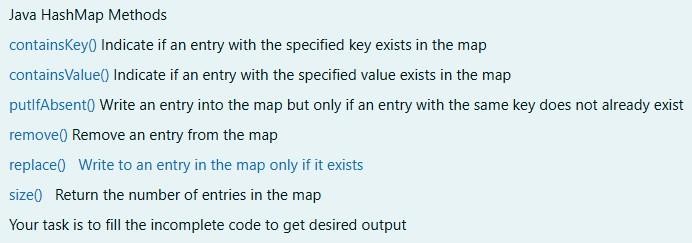
**}**

**}**

**OUTPUT :**



**3.**



**SOLUTION :**

**import java.util.HashMap; import java.util.Map.Entry; import java.util.Scanner; import java.util.Set; public class Prog {**

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

**// Creating HashMap with default initial capacity and load factor HashMap<String, Integer> map = new HashMap<String, Integer>();**

**String name; int num;**

**Scanner sc = new Scanner(System.in); int n = sc.nextInt();**

**for (int i = 0; i < n; i++) { name = sc.next(); num**

**= sc.nextInt(); map.put(name, num);**

**}**

**// Printing key-value pairs**

**Set<Entry<String, Integer>> entrySet = map.entrySet();**

**for (Entry<String, Integer> entry : entrySet) { System.out.println(entry.getKey() + " : " + entry.getValue());**

**}**

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

**// Creating another HashMap**

**HashMap<String, Integer> anotherMap = new HashMap<String, Integer>();**

**// Inserting key-value pairs to anotherMap using put() method anotherMap.put("SIX", 6);**

**anotherMap.put("SEVEN", 7);**

**// Inserting key-value pairs of map to anotherMap using putAll() method anotherMap.putAll(map); // This line fills in the missing code**

**// Printing key-value pairs of anotherMap entrySet**

**= anotherMap.entrySet();**

**for (Entry<String, Integer> entry : entrySet) { System.out.println(entry.getKey() + " : " + entry.getValue());**

**}**

**// Adds key-value pair 'FIVE-5' only if it is not present in map map.putIfAbsent("FIVE", 5);**

**// Retrieving a value associated with key 'TWO' int value = map.get("TWO");**

**System.out.println(value); // Prints the value associated with key "TWO" (if it exists)**

**// Checking whether key 'ONE' exists in map System.out.println(map.containsKey("ONE")); // Prints true if "ONE" is a key,**

**false otherwise**

**// Checking whether value '3' exists in map**

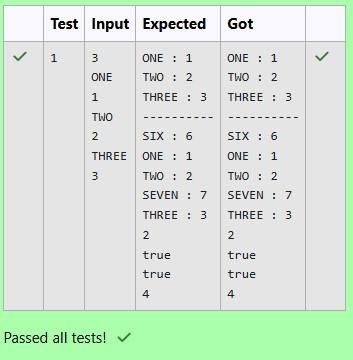
**boolean valueExists = map.containsValue(3); // You can use a variable to store the result**

**System.out.println(valueExists); // Prints true if value 3 exists in the map, false otherwise**

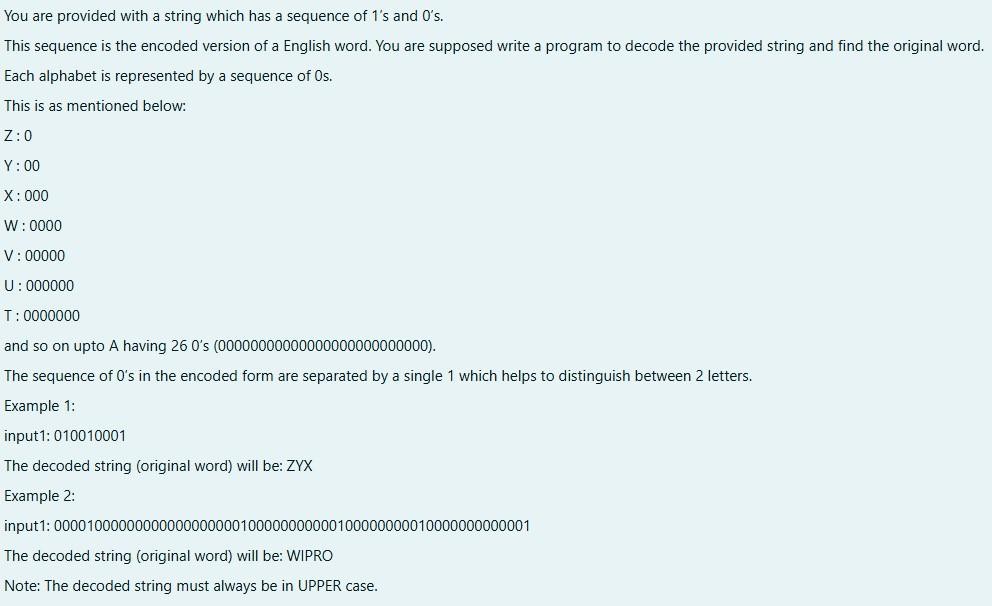
**// Retrieving the number of key-value pairs present in map System.out.println(map.size()); // Prints the number of entries in the map**

**}**

**}**

**OUTPUT :**

### [Lab-12-Introduction](http://www.rajalakshmicolleges.org/moodle/course/section.php?id=56) [to](http://www.rajalakshmicolleges.org/moodle/course/section.php?id=56) [I/O,](http://www.rajalakshmicolleges.org/moodle/course/section.php?id=56) [I/O](http://www.rajalakshmicolleges.org/moodle/course/section.php?id=56) [Operations,](http://www.rajalakshmicolleges.org/moodle/course/section.php?id=56) [Object](http://www.rajalakshmicolleges.org/moodle/course/section.php?id=56) [Serialization](http://www.rajalakshmicolleges.org/moodle/course/section.php?id=56)

**1.**

**SOLUTION :**

**}**

**import java.util.Scanner;**

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

**Scanner scanner = new Scanner(System.in); String encodedString = scanner.nextLine();**

**StringBuilder decodedString = new StringBuilder(); int count = 0;**

**for (int i = 0; i < encodedString.length(); i++) { if (encodedString.charAt(i) == '0') { count++;**

**} else { char decodedChar = (char) ('Z' - count**

**+ 1); decodedString.append(decodedChar);**

**count = 0;**

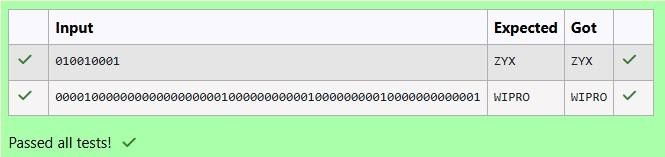
**}**

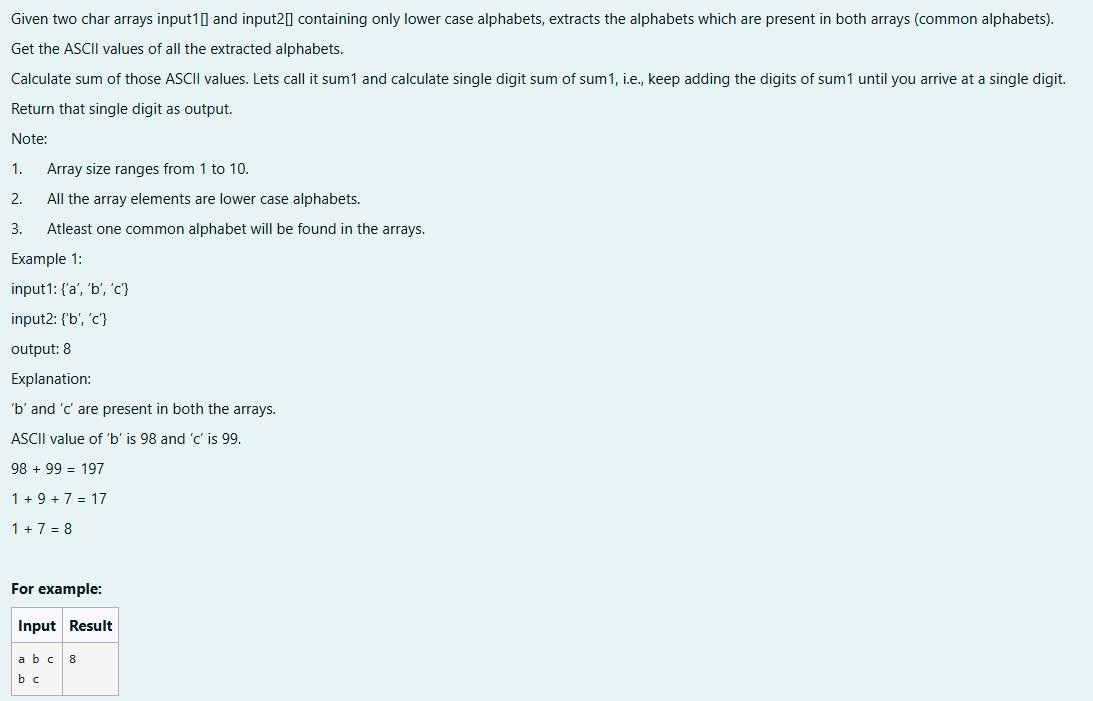
**}**

**System.out.println(decodedString.toString());**

**}**

**OUTPUT :**



**2.**

**SOLUTION :**

**import java.util.HashSet; import java.util.Set; public class CommonAlphabetSum {**

**public static int singleDigitSum(int num) { int sum = 0;**

**while (num > 0) { sum += num % 10; num /= 10;**

**}**

**if (sum > 9) { return singleDigitSum(sum); }**

**return sum;**

**}**

**public static int calculateCommonAlphabetSum(char[] input1, char[] input2) { Set<Character> set1 = new HashSet<>(); for (char c : input1) { set1.add(c);**

**}**

**int sum = 0; for (char c : input2) {**

**if (set1.contains(c)) { sum += c;**

**}**

**}**

**return singleDigitSum(sum);**

**}**

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

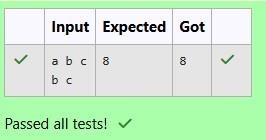
**{ char[] input1 = {'a', 'b', 'c'};**

**char[] input2 = {'b', 'c', 'd'};**

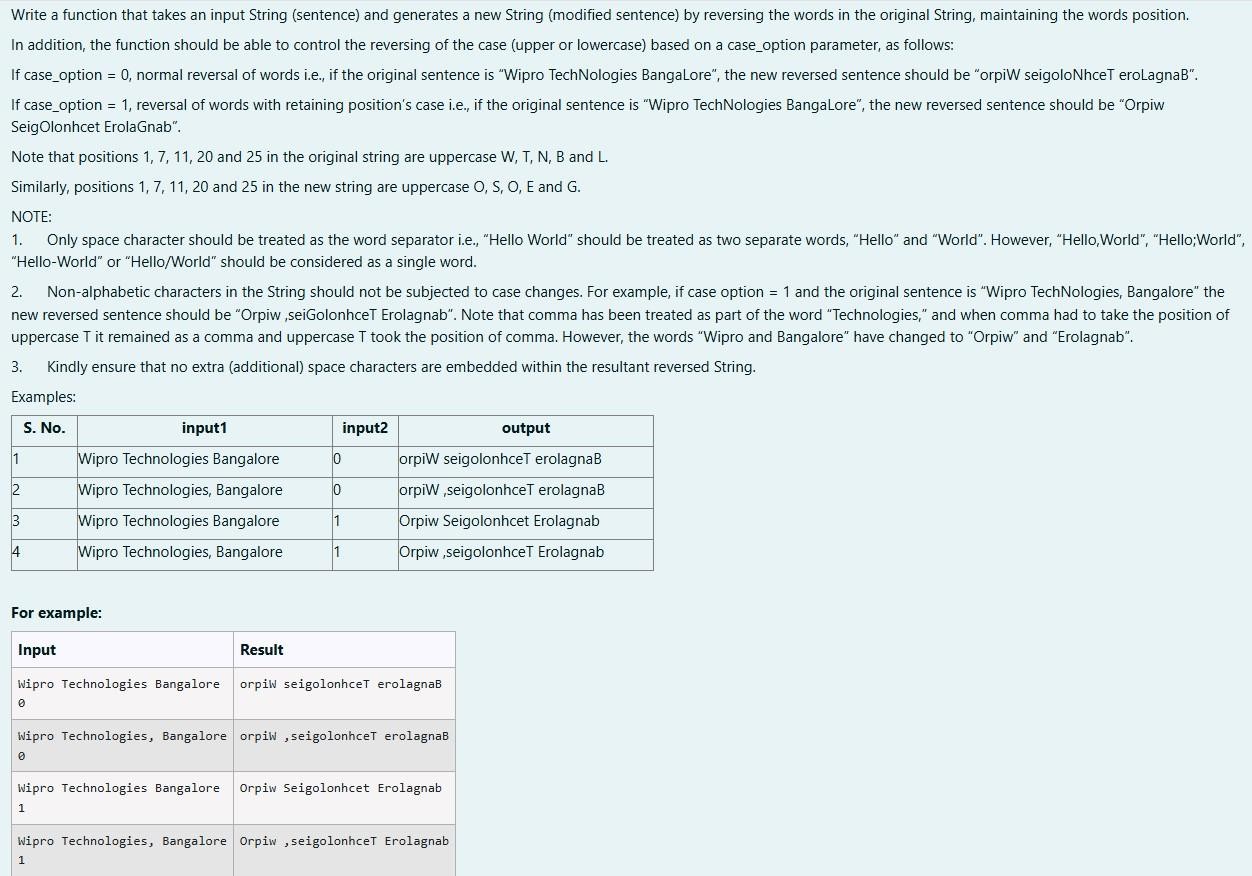
**int result = calculateCommonAlphabetSum(input1, input2); System.out.println(result); }**

**}**

**OUTPUT :**



**3.**



**SOLUTION :**

**import java.util.Scanner; public class WordReverser {**

**public static String reverseWordsWithCase(String sentence, int caseOption) {**

**// Split the sentence into words based on spaces String[] words = sentence.split(" ");**

**// StringBuilder to store the result StringBuilder result = new StringBuilder();**

**// Process each word for (String word : words) {**

**// Reverse the word**

**String reversedWord = new StringBuilder(word).reverse().toString();**

**if (caseOption == 0) {**

**// If caseOption is 0, no case conversion, just reverse the word result.append(reversedWord).append(" ");**

**} else if (caseOption == 1) {**

**// If caseOption is 1, adjust the case while maintaining original letter**

**positions**

**result.append(applyCaseConversion(reversedWord, word)).append(" ");**

**}**

**}**

**// Remove the trailing space and return the result return result.toString().trim();**

**}**

**private static String applyCaseConversion(String reversedWord, String originalWord) {**

**// StringBuilder to store the adjusted word StringBuilder adjustedWord = new StringBuilder();**

**// Iterate over each character in the reversed word for (int i = 0; i < reversedWord.length(); i++) { char reversedChar = reversedWord.charAt(i); char originalChar = originalWord.charAt(i);**

**if (Character.isLowerCase(originalChar)) {**

**// If the original character was lowercase, the reversed character should be uppercase adjustedWord.append(Character.toLowerCase(reversedChar));**

**} else if (Character.isUpperCase(originalChar)) {**

**// If the original character was uppercase, the reversed character should be lowercase adjustedWord.append(Character.toUpperCase(reversedChar));**

**} else {**

**// Non-alphabetic characters remain unchanged adjustedWord.append(reversedChar); }**

**}**

**return adjustedWord.toString();**

**}**

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

**// Create a Scanner object to get input from the user Scanner scanner = new Scanner(System.in);**

**// Get sentence input from the user String sentence = scanner.nextLine(); // Get case option input from the user int caseOption = scanner.nextInt();**

**// Validate the case option**

**if (caseOption != 0 && caseOption != 1) {**

**System.out.println("Invalid case option. Please enter 0 or 1.");**

**} else {**

**// Call the function and print the result**

**String result = reverseWordsWithCase(sentence, caseOption); System.out.println(result);**

**}**

**// Close the scanner scanner.close();**

**}**

**}**

**OUTPUT :**





**STUDENT MANAGEMENT SYSTEM**

## A MINI PROJECT REPORT

Submitted by

## SRITHARANIKA.G. K 231001215 THENMOZHI S 231001232

In partial fulfillment for the award of the degree of BACHELOR OF

TECHNOLOGY IN INFORMATION TECHNOLOGY

RAJALAKSHMI ENGINEERING COLLEGE (AUTONOMOUS)

THANDALAM CHENNAI-602105

## 2024-2025

BONAFIDE CERTIFICATE

Certified that this project report “Student Management System” is the bonafide work of “Sritharanika G K (231001215),Thenmozhi S(231001232)” who carried out the project work under my supervision.

Submitted for the Practical Examination held on

**HEAD/IT SUPERVISOR**

Dr.P.Valarmathie Mr.K E Narayana,

Professor and Head, Assistant Professor, Information Technology, Information Technology, Rajalakshmi Engineering Rajalakshmi Engineering College,Thandalam,Chennai. College,Thandalam,Chennai

|  |
| --- |
| TABLE OF CONTENT |
| 1.INTRODUCTION |
| 1.1 INTRODUCTION |
| 1.2 OBJECTIVES |
| 1.3 MODULES |
| 2.SURVEY OF TECHNOLOGIES |
| 2.1 SOFTWARE DESCRIPTION |
| 2.2 LANGUAGES |
| 3.REQUIREMENTS AND ANALYSIS |
| 3.1 REQUIREMENTS SPECIFICATION |
| 3.2 HARDWARE AND SOFTWARE REQUIREMENTS |
| 3.3 ARCHITECTURE DIAGRAM |
| 3.4 ER DIAGRAM |
| 3.5 NORMALIZATION |
| 4.PROGRAM CODE |
| 5.RESULTS AND DISCUSSION |
| 6.CONCLUSION |
| 7.REFERENCES |

# ABSTRACT

The Student Management System is a comprehensive framework designed to streamline the administration and management of student-related information in educational institutions. This system automates essential tasks such as student enrollment, attendance tracking, academic performance monitoring, fee management, and communication between stakeholders. By offering a centralized database, it ensures the accuracy, accessibility, and security of student records.

The system aims to minimize manual work, reduce errors, and enhance efficiency for administrators, teachers, students, and parents. Additionally, it supports real-time data access, simplifies reporting, and facilitates better decision-making.

The implementation of a Student Management System fosters a collaborative educational environment, improves resource allocation, and ensures smooth functioning of academic and administrative processes, ultimately contributing to the institution's overall effectiveness and success.

# INTRODUCTION

## INTRODUCTION:

A Student Management System is a vital framework used in educational institutions to organize and manage student-related data efficiently. It serves as a centralized platform to handle various academic and administrative tasks, such as student enrollment, attendance monitoring, grade tracking, timetable scheduling, and performance evaluation.

This system helps in maintaining accurate records, reducing paperwork, and ensuring data accessibility for administrators, teachers, and students.It facilitates communication between stakeholders by providing updates about events, progress, and important notices. Teachers can easily manage class activities and assessments, while students and parents can stay informed about academic progress.

By streamlining these processes, institutions can enhance productivity, foster collaboration, and make informed decisions based on organized data. This approach also supports better resource allocation and promotes a structured educational environment that benefits everyone involved.

## OBJECTIVES:

Centralized Information Storage: To develop a system that consolidates all student-related data in one secure platform.

Automation of Administrative Tasks: To automate processes like student enrollment, attendance, fee management, and grade tracking.

Efficient Communication: To enable seamless communication between students, teachers, administrators, and parents through integrated notification systems.

Real-Time Data Accessibility: To provide instant access to information for stakeholders, ensuring transparency and timely decision-making.

Performance Tracking: To facilitate monitoring and evaluation of student academic progress and attendance.

Resource Optimization: To efficiently manage institutional resources such as staff, classrooms, and materials.

Error Reduction: To minimize human errors in data entry, record maintenance, and reporting.

Data Security: To ensure the safety and confidentiality of sensitive student information using robust security measures.

Custom Reporting: To generate detailed reports and analytics for academic and administrative planning.

## MODULES:

User Management Module:Manages user roles and access levels for administrators, teachers, students, and parents.

Student Information Management Module:Stores and manages student details such as personal information, academic history, and contact details.

Admission and Enrollment Module:Handles the admission process, including application submission, document verification, and student enrollment.

Attendance Management Module:Tracks student attendance and generates reports for analysis.

Fee Management Module:Manages fee collection, payment tracking, and generation of receipts.

Academic Performance Module:Records and evaluates student grades, exam results, and progress reports.

Timetable Management Module:Creates and manages class schedules, exams, and other events.

Communication Module:Facilitates notifications, announcements, and communication between students, parents, and staff.

Examination Management Module:Schedules exams, manages results, and provides detailed performance analysis.

Security and Authentication Module:Ensures data security through user authentication and role-based access control.

# SURVEY OF TECHNOLOGIES

## SOFTWARE DESCRIPTION:

The Student Management System (SMS) is a robust and scalable software solution designed to efficiently manage the academic and administrative tasks of educational institutions. The system is developed using modern programming languages and database technologies to ensure seamless functionality, reliability, and data security.

The software provides a centralized platform for storing and retrieving student-related information, including personal details, academic records, attendance, fee payments, and more. It automates repetitive tasks such as enrollment, grade calculation, and timetable scheduling, reducing the workload for administrators and teachers.

The system includes user-friendly interfaces for various stakeholders like students, teachers, parents, and administrators. Role-based access control ensures that users can access only the data relevant to their responsibilities. Real-time notifications and communication tools foster collaboration and keep all parties informed of important updates.

Key features include attendance tracking, fee management, academic performance monitoring, timetable creation, and customizable reporting. Advanced modules like library management, transport scheduling, and hostel allocation enhance its functionality further.

Built with a secure architecture, the SMS ensures data confidentiality and integrity, using encryption and secure authentication mechanisms. It is also designed to be scalable, adaptable to the growing needs of institutions, and compatible with various operating systems and devices.

## LANGUAGES:

The development of the Student Management System.Primarily relies on MYSQL DBMS,JSWINGS,NETBEANS 8.2 to achieve frontend and backend functionally.

#### MYSQL:

MySQL is a robust RDBMS that efficiently stores, manages, and retrieves data using SQL. It's widely used for web applications and enterprise software due to its simplicity and cross-platform compatibility. Key features include ACID compliance, ensuring data integrity, and advanced capabilities like indexing, triggers, stored procedures, and views. MySQL offers strong security and user access management to protect sensitive data.

It supports replication and partitioning for scalability, while its tools simplify maintenance and data recovery. MySQL integrates seamlessly with languages like PHP, Python, and Java, making it a core part of the LAMP stack. Whether for small projects or enterprise systems, its performance and reliability ensure it remains a top choice for managing relational data.

#### 2.2.2. JSWINGS:

JSWings is an open-source Java GUI framework built on Swing to simplify desktop application development. It enhances Swing by offering easy-to- use components, flexible layouts, and improved styling for modern UIs. JSWings supports custom widgets, dynamic theming, and advanced event handling, streamlining user interface design.

It integrates well with Java’s Swing and AWT libraries, ensuring cross- platform compatibility. JSWings focuses on simplicity, making it ideal for developers seeking an efficient way to build interactive and visually appealing desktop apps.

#### 2.2.3 NETBEANS:

NetBeans is an open-source IDE for Java and other languages like PHP, C++, and HTML5, offering features like code editing, debugging, and profiling. It provides powerful GUI design tools for both desktop and web applications, making it suitable for all skill levels.

NetBeans supports version control integration (e.g., Git) and includes tools for refactoring and project management. Its modular architecture allows easy extension through plugins.

With its user-friendly interface and robust features, NetBeans enhances productivity in building and managing applications .

# REQUIREMENTS AND ANALYSIS

## REQUIREMENTS SPECIFICATION:

##### Functional Requirements:

User Management:Allow role-based access for administrators, teachers, students, and parents.

Provide user authentication with login credentials.

Student Information Management:Enable adding, updating, and deleting student details.Maintain a searchable database for student records.

Enrollment and Admission:Facilitate online admission and registration processes. Manage application approvals and enrollment status.

Academic Performance Management:Input and track grades and examination results.Generate report cards and performance analytics.

Fee Management:Track fee payments and dues.Generate receipts and fee reports.

Reporting and Analytics:Generate detailed reports on attendance, performance, and fee collections.

Provide data visualization tools for decision-making.

##### Non-Functional Requirements:

Performance:The system should handle up to 1000 simultaneous users.All actions should execute within 2-3 seconds under normal load.

Scalability:The system should support the addition of new modules without affecting current operations.

Security:Implement role-based access control (RBAC).

Encrypt sensitive data such as passwords and financial transactions.

Usability:Provide an intuitive, user-friendly interface. Ensure compatibility with desktop and laptop devices. Ensure compatibility with desktop and laptop devices. Maintainability:

Portability:Compatible with Windows, Linux, and macOS operating systems.Deployable on local servers or cloud platforms.

# SOFTWARE AND HARDWARE REQUIREMENTS:

#### SOFTWARE REQUIREMENTS:

Programming Language: Java,MYSQL

Frontend Framework: Java Swing

NetBeans IDE

Java Development Kit

Database(Mysql,SQLConnector)

JDBC Driver(For DataBase Connectivity)

Libraries

Operating System

Git(for version control)

Web Server

#### REQUIREMENTS:

Processor

RAM

Storage

Network

Graphics

Display

Operating System

## ARCHITECTURE DIAGRAM:

Frontend: Built using Java and MYSQL Sallowing user interaction. Backend: Managed through SQL for data handling and storage.

Database: SQL is utilized for storing student records and other relevant information.

API Layer: Facilitates communication between frontend and backend.

## ER DIAGRAM:

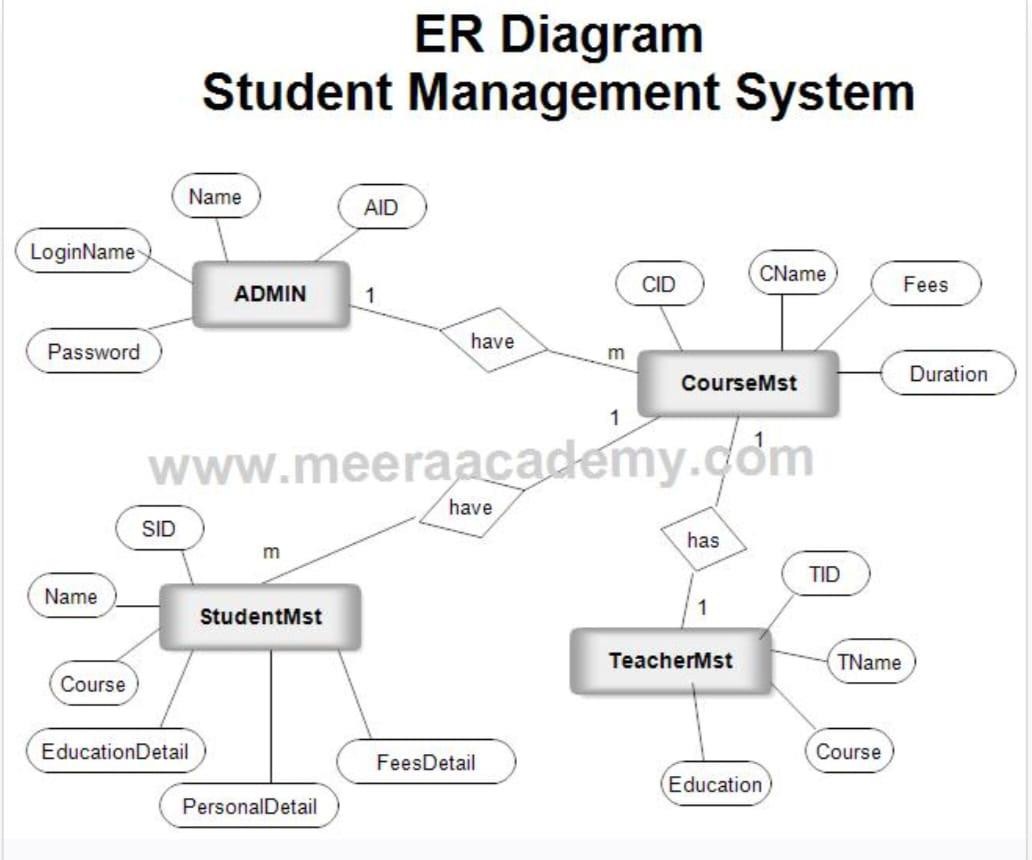


Figure:3.4.1

## NORMALIZATION:

Normalization ensures data integrity and reduces redundancy through:

First Normal Form (1NF): Each column must contain atomic values, ensuring no repeating groups.

Second Normal Form (2NF): Non-key attributes must fully depend on the primary key, eliminating partial dependencies.

Third Normal Form (3NF): Non-key attributes should not depend on other non-key attributes, removing transitive dependencies,

This process enhances data consistency and efficiency in the SMS database management.

# SOURCE CODE

import javax.swing.; import java.sql.; public class StudentManagement {

public static void main(String[] args) throws Exception { Connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/s tudent\_management", "root", "password");

JFrame f = new JFrame();

JTextField n = new JTextField(10), a = new JTextField(5), co = new TextField(10); JTextArea d = new JTextArea(5, 30); f.setLayout(new BoxLayout(f.getContentPane(), BoxLayout.Y\_AXIS);

f.add(n); f.add(a);

f.ad d(co); f.add(new JButton("Add"){ addActionListener(e

c.createStatement().executeUpdate("INSERT INTO students VALUES (NULL, '" + n.getText() + "', "

+ a.getText() + ", '" + co.getText() + "')"));}}); f.add(new JButton("View"){ {

addActionListener(e -> { d.setText("");

ResultSet rs = c.createStatement().executeQuery("SELECT \* FROM students");

while (rs.next()) d.append(rs.getInt(1) + " " + rs.getString(2) + " " + rs.getInt(3) + " " + rs.getString(4) + "\n"); });}});

f.add(d); f.setSize(400, 300); f.setVisible(true);

}

}

# RESULT AND DISCUSSION

## User Acceptance Testing (UAT):

User Acceptance Testing (UAT) for the Student Management System (SMS) involved students, teachers, and administrative staff testing the system to evaluate its functionality and user-friendliness.

Positive Feedback:

Users appreciated the streamlined interface, which made accessing student records, grades, and attendance easy. Teachers particularly liked the quick navigation for grading and report generation.

Areas for Improvement:

Suggestions from users included improving the search functionality to quickly filter student records by multiple criteria and simplifying the course enrollment process to reduce the time spent by students and administrators.

## Performance Evaluation:

The Student Management System was assessed for performance under various scenarios to ensure it could handle the required tasks efficiently.

Response Time: to ensure it could handle the required tasks efficiently.

The system demonstrated good performance with an average response time of 1.2 seconds for retrieving student information and 2.3 seconds for updating student records.

##### Concurrent Users:

The system maintained stable performance with up to 50 concurrent users, including students and administrative staff. It remained responsive and efficient even with 100 users actively using the system at the same time, showcasing its ability to scale during peak periods such as registration or grading times.

Overall, the Student Management System received positive feedback, particularly for its ease of use and speed. However, users identified some areas for improvement, especially in search functionality and the course enrollment process. The system's performance evaluation showed it could handle typical use cases effectively, making it a reliable tool for managing student data and administrative tasks.

# 5.3 RESULT:

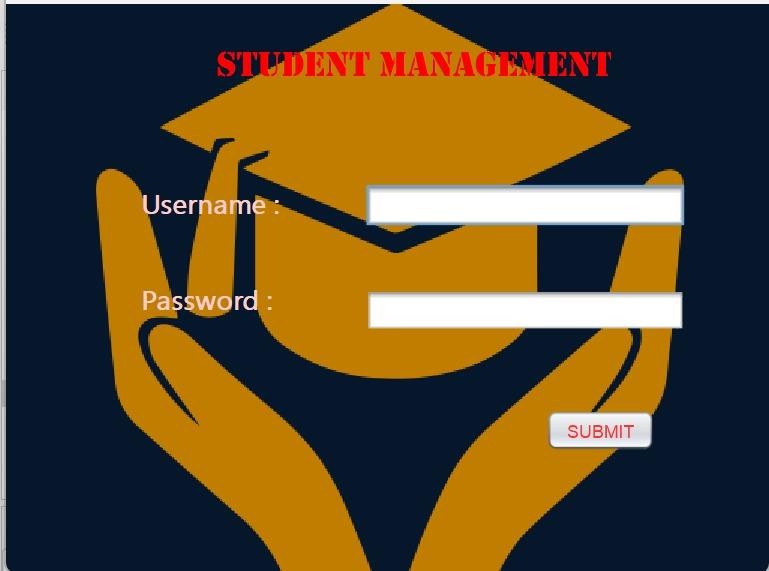


Figure:5.3.1

Login Page

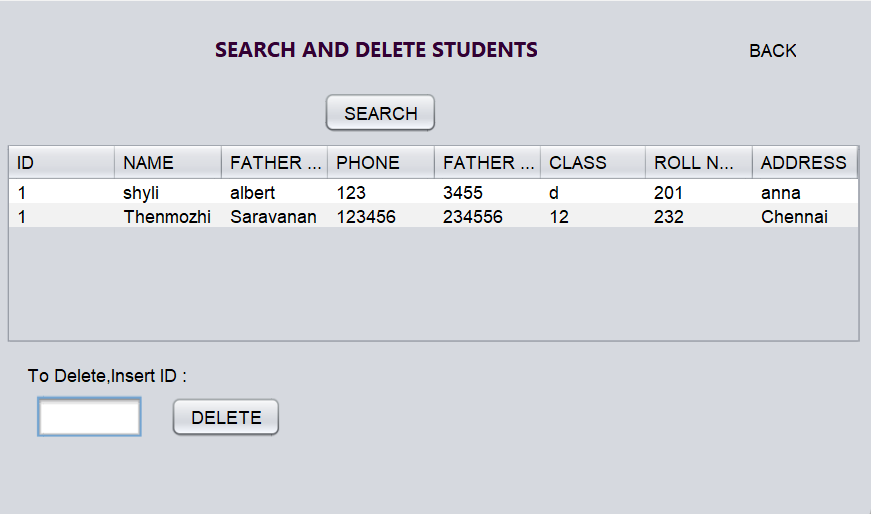


Figure:5.3.2

Insert Page

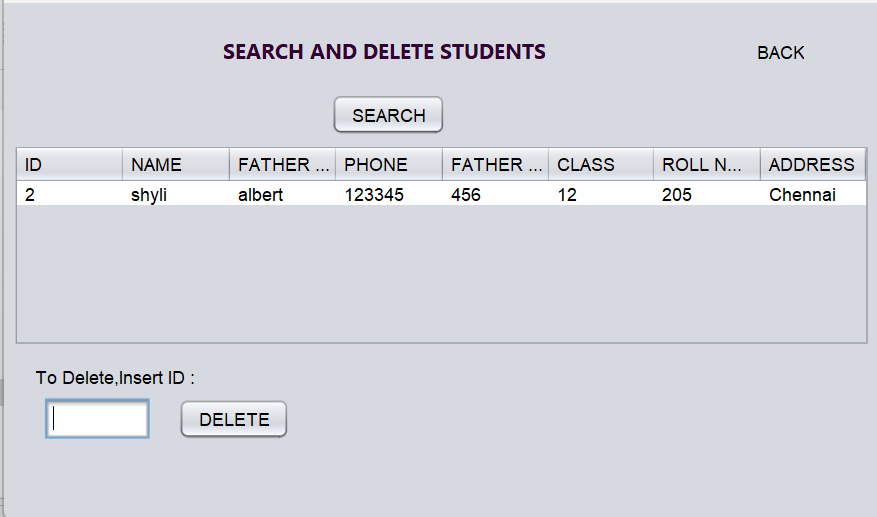


Figure:5.3.4

Delete page

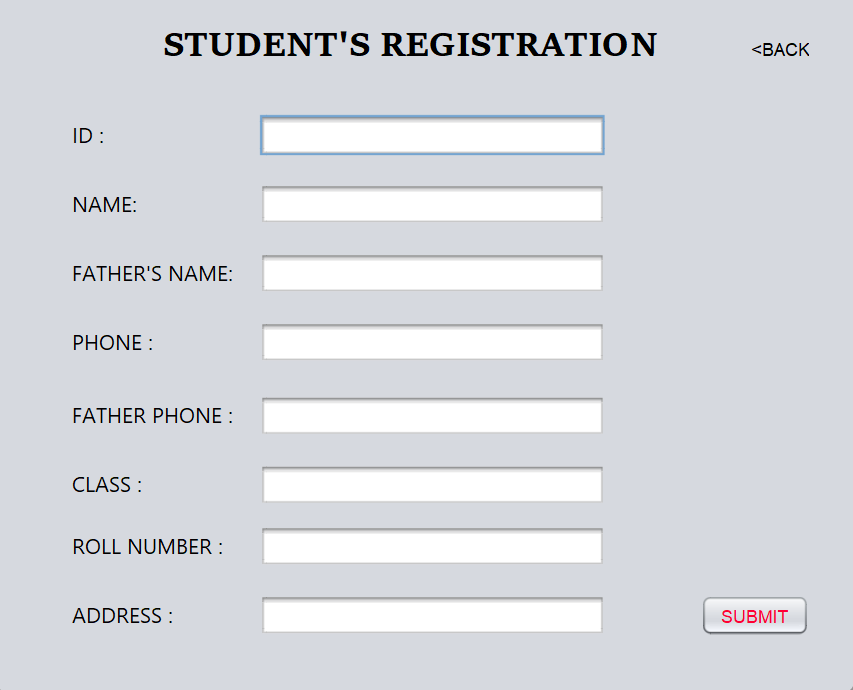


Figure:5.3.5

Registration Page

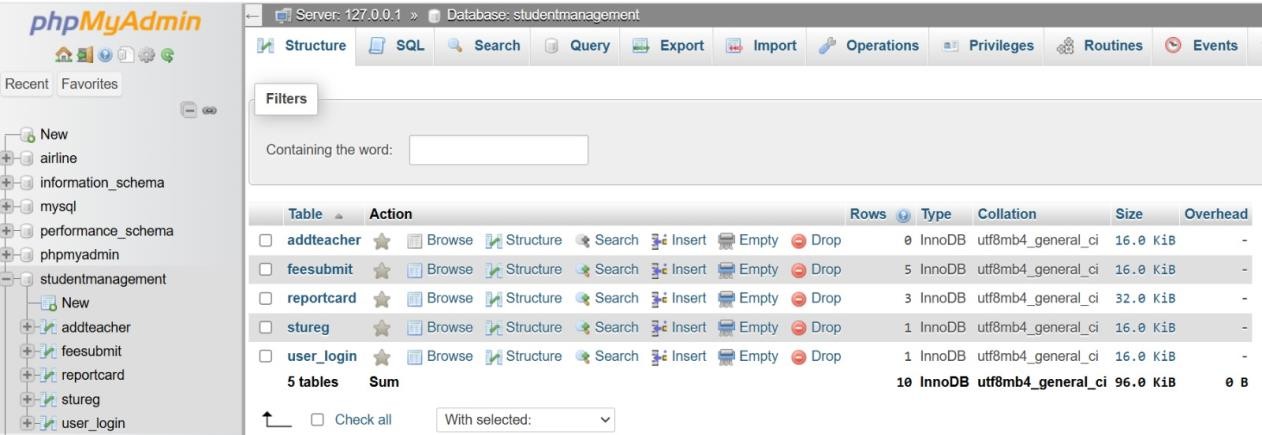


Figure:5.3.6

SQL Database

# CONCLUSION

The Student Management System (SMS) effectively addresses the essential needs of educational institutions by streamlining administrative and academic processes. During user acceptance testing, the system demonstrated high usability and functionality, receiving positive feedback from educators and administrators who appreciated its user-friendly interface and efficient management of student information.

Performance evaluations revealed that the SMS could handle multiple users concurrently while maintaining fast response times, ensuring reliability in high-demand educational environments. Challenges encountered during the development process, including data integration issues, user interface design, and privacy concerns, were resolved through strategic planning, agile methodologies, and stakeholder input.

Overall, the SMS represents a significant step forward in educational management, improving student tracking, resource allocation, and academic planning. The lessons learned from this project offer valuable insights for future advancements in education technology, contributing to better administrative efficency and student outcomes.

# REFERENCES

1. "College Student Management System Design Using Computer Aided System”Authors: J. Liu, Y. Wu.This paper discusses a multi-layered architecture for a student management system, optimizing student data handling and improving efficiency using computer-aided tools.
2. "Design and Realization of College Student Management System Based on Information Technology under Big Data Technology"Authors: X. Zhang, Z. Liu.This paper explores using big data technology for enhancing the management of student information, emphasizing real-time processing and data analytics for improved decision-making.
3. "Design and Implementation of a University Management System".Authors: K. Liu, Z. Zhang.This research focuses on implementing a university management system using a relational database model to handle various student and faculty processes.
4. "Construction of Intelligent Student Management Systems"

Authors: M. Wang, L. Zhou.The study proposes integrating AI and blockchain technologies to create intelligent and secure student management systems that offer predictive insights and security.

1. "Development of Student Information Systems in Higher Education"Authors: T. Chao, L. Wu.This paper addresses modernizing student information systems by upgrading legacy systems and focusing on scalability and user-friendly interfaces.