**Practical No. 1:**

Write a java program to take input as a command line argument. Your name, course, universityrollno and semester. Display the information.

Name:

University Roll no:

Course:

Semester:

**Source Code:**

public class q1 {

public static void main(String args[]) {

if (args.length < 4) {

System.out.println("Invalid arguments");

return;

}

String name = args[0];

String universityRollNo = args[1];

String course = args[2];

String semester = args[3];

System.out.println("Name: " + name);

System.out.println("University Roll no: " + universityRollNo);

System.out.println("Course: " + course);

System.out.println("Semester: " + semester);

}

}

**Output:**

PS C:\FOLDER\Programming\java\Sarthak Kothiyal A2 60> javac q1.java

PS C:\FOLDER\Programming\java\Sarthak Kothiyal A2 60> java q1 Sarthak 2319517 B.Tech 4

Name: Sarthak

University Roll no: 2319517

Course: B.Tech

Semester: 4

PS C:\FOLDER\Programming\java\Sarthak Kothiyal A2 60>

**Practical No. 2:**

Using the switch statement, write a menu-driven program to calculate the maturity amount of a bank deposit.

The user is (i) Term Deposit (ii) Recurring Deposit

For option (i) accept Principal (p), rate of interest (r) and time period in years (n). Calculate and output the maturity amount (a) receivable using the formula a = p[1 + r / 100]n.

For option (ii) accept monthly installment (p), rate of interest (r) and time period in months (n). Calculate and output the maturity amount (a) receivable using the formula a = p\*n+p\*n(n + 1) / 2 \* r / 100 \* 1 / 12. For an incorrect option, an appropriate error message should be displayed.

**Source Code:**

import java.text.DecimalFormat;

import java.util.Scanner;

public class q2 {

public static void main(String args[]) {

Scanner sc = new Scanner(System.in);

System.out.println("1. Term Deposit\n2. Recurring Deposit\nEnter the choice: ");

int ch = sc.nextInt();

double p, r, t, amount;

DecimalFormat d = new DecimalFormat("#.##");

switch (ch) {

case 1:{

System.out.println("Please enter principal:");

p = sc.nextDouble();

System.out.println("Please enter rate:");

r = sc.nextDouble();

System.out.println("Please enter time period in years:");

t = sc.nextDouble();

amount = p \* Math.pow((1 + (r / 100)), t);

System.out.println("Total amount after term deposit: " + d.format(amount));

break;

}

case 2:{

System.out.println("Please enter monthly installment amount:");

p = sc.nextDouble();

System.out.println("Please enter rate:");

r = sc.nextDouble();

System.out.println("Please enter time period in months:");

t = sc.nextDouble();

amount = (p \* t) + (p \* (t \* (t + 1)) / 2 \* (r / 100) \* (1.0 / 12));

System.out.println("Total amount after recurring deposit: " + d.format(amount));

break;

}

default:{

System.out.println("Enter a valid option.");

}

}

}

}

**Output:**

PS C:\FOLDER\Programming\java\Sarthak Kothiyal A2 60> javac q2.java

PS C:\FOLDER\Programming\java\Sarthak Kothiyal A2 60> java q2

1. Term Deposit

2. Recurring Deposit

Enter the choice:

1

Please enter principal:

10000

Please enter rate:

5

Please enter time period in years:

3

Total amount after term deposit: 11576.25

PS C:\FOLDER\Programming\java\Sarthak Kothiyal A2 60>

PS C:\FOLDER\Programming\java\Sarthak Kothiyal A2 60> javac q2.java

PS C:\FOLDER\Programming\java\Sarthak Kothiyal A2 60> java q2

1. Term Deposit

2. Recurring Deposit

Enter the choice:

2

Please enter monthly installment amount:

15000

Please enter rate:

6

Please enter time period in months:

4

Total amount after recurring deposit: 60750

PS C:\FOLDER\Programming\java\Sarthak Kothiyal A2 60>

**Practical No. 3:**

Program to find if the given numbers are Friendly pair or not (Amicable or not). Friendly Pair are two or more numbers with a common abundance.

**Source Code:**

import java.util.Scanner;

public class q3 {

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

System.out.println("Enter the number to check: ");

int num=sc.nextInt(),sum=0;

for(int i=1;i<=num/2;i++){

if(num%i==0){

sum=sum+i;

}

}

if(sum==num){

System.out.println(num+" is a Friendly Number...");

}

else{

System.out.println(num+" is Not a Friendly Number...");

}

}

}

**Output:**

PS C:\FOLDER\Programming\java\Sarthak Kothiyal A2 60> javac q3.java

PS C:\FOLDER\Programming\java\Sarthak Kothiyal A2 60> java q3

Enter the number to check:

99

99 is Not a Friendly Number...

PS C:\FOLDER\Programming\java\Sarthak Kothiyal A2 60>

PS C:\FOLDER\Programming\java\Sarthak Kothiyal A2 60> javac q3.java

PS C:\FOLDER\Programming\java\Sarthak Kothiyal A2 60> java q3

Enter the number to check:

28

28 is a Friendly Number...

PS C:\FOLDER\Programming\java\Sarthak Kothiyal A2 60>

**Practical No. 4:**

Program to replace all 0's with 1 in a given integer. Given an integer as an input, all the 0's in the number has to be replaced with 1.

**Source Code:**

import java.util.Scanner;

public class q4 {

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

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

int num=sc.nextInt();

int temp=num,pv=0;

while(temp!=0){

if(temp%10==0)

num=num+(int)Math.pow(10,pv);

pv++;

temp=temp/10;

}

System.out.println("Updated Number: "+num);

}

}

**Output:**

PS C:\FOLDER\Programming\java\Sarthak Kothiyal A2 60> javac q4.java

PS C:\FOLDER\Programming\java\Sarthak Kothiyal A2 60> java q4

Enter the number:

10500609

Updated Number: 11511619

PS C:\FOLDER\Programming\java\Sarthak Kothiyal A2 60>

PS C:\FOLDER\Programming\java\Sarthak Kothiyal A2 60> javac q4.java

PS C:\FOLDER\Programming\java\Sarthak Kothiyal A2 60> java q4

Enter the number:

506040000

Updated Number: 516141111

PS C:\FOLDER\Programming\java\Sarthak Kothiyal A2 60>

**Practical No. 5:**

Printing an array into Zigzag fashion. Suppose you were given an array of integers, and you are told to sort the integers in a zigzag pattern. In general, in a zigzag pattern, the first integer is less than the second integer, which is greater than the third integer, which is less than the fourth integer, and so on. Hence, the converted array should be in the form of e1 < e2 > e3 < e4 > e5 < e6.

**Source Code:**

import java.util.Scanner;

public class q5 {

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

System.out.println("Enter size of array: ");

int n=sc.nextInt();

int arr[]=new int[n];

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

arr[i]=sc.nextInt();

}

for(int i=0;i<n-1;i++){

if(arr[i]>arr[i+1]){

int temp=arr[i];

arr[i]=arr[i+1];

arr[i+1]=temp;

}

i++;

if(arr[i]<arr[i+1]){

int temp=arr[i];

arr[i]=arr[i+1];

arr[i+1]=temp;

}

}

System.out.println("Updated Array: ");

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

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

}

}

}

**Output:**

PS C:\FOLDER\Programming\java\Sarthak Kothiyal A2 60> javac q5.java

PS C:\FOLDER\Programming\java\Sarthak Kothiyal A2 60> java q5

Enter size of array:

7

4 3 7 8 6 2 1

Updated Array:

3 7 4 8 2 6 1

PS C:\FOLDER\Programming\java\Sarthak Kothiyal A2 60>

PS C:\FOLDER\Programming\java\Sarthak Kothiyal A2 60> javac q5.java

PS C:\FOLDER\Programming\java\Sarthak Kothiyal A2 60> java q5

Enter size of array:

5

1 2 3 4 5

Updated Array:

1 3 2 5 4

PS C:\FOLDER\Programming\java\Sarthak Kothiyal A2 60>

**Practical No. 6**

The problem to rearrange positive and negative numbers in an array.  
  
**Source Code:**

import java.util.Scanner;

public class positiveNegativeArray {

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

System.out.println("Enter the size of array: ");

int n=sc.nextInt();

int arr[]=new int[n];

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

arr[i]=sc.nextInt();

}

int low=0,high=n-1;

while(low<high){

if(arr[low]>=0 && arr[high]<0){

int temp=arr[high];

arr[high]=arr[low];

arr[low]=temp;

low++;

high--;

}

else if(arr[low]<0){

low++;

}

else if(arr[high]>=0){

high--;

}

}

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

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

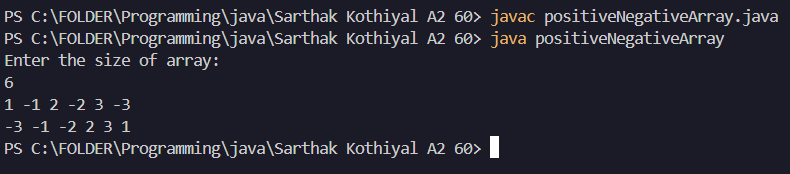
}

sc.close();

}

}

**OUTPUT:**



**Practical No. 7**Program to find the saddle point coordinates in a given matrix. A saddle point is an element of the matrix, which is the minimum element in its row and the maximum in its column.

**Source Code:**

import java.util.Scanner;

public class saddleElement {

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

int m,n;

System.out.println("Enter size of Matrix: ");

m=sc.nextInt();

n=sc.nextInt();

int arr[][]=new int[m][n];

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

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

arr[i][j]=sc.nextInt();

}

}

boolean flag=false;

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

int min=arr[i][0];

int minIndex=0;

for(int j=1;j<n;j++){

if(arr[i][j]<min){

min=arr[i][j];

minIndex=j;

}

}

boolean found=true;

for(int k=0;k<m;k++){

if(arr[k][minIndex]>arr[i][minIndex]){

found=false;

break;

}

}

if(found){

System.out.println("Saddle Point found: "+arr[i][minIndex]);

flag=true;

}

}

if(!flag){

System.out.println("No Saddle Point found...");

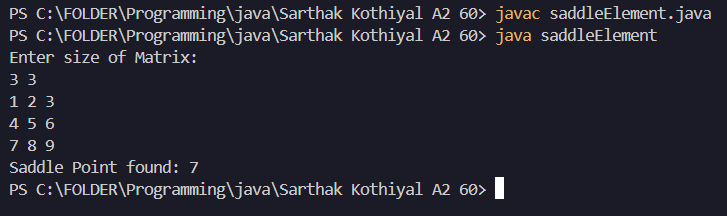
}

sc.close();

}

}

**OUTPUT:**



**Practical No. 8**

Program to find all the patterns of 0(1+)0 in the given string. Given a string containing 0's and 1's, find the total number of 0(1+)0 patterns in the string and output it. 0(1+)0 - There should be at least one '1' between the two 0's.

**Source Code:**

import java.util.Scanner;

public class pattern {

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

System.out.print("Enter a Binary String: ");

String str=sc.nextLine();

int count=0;

int i=0,left,right=0;

while(i<str.length()){

if(str.charAt(i)=='0'){

left=i;

i++;

while(i<str.length() && str.charAt(i)=='1'){

i++;

}

if(i<str.length() && str.charAt(i)=='0'){

right=i;

count++;

System.out.println(str.substring(left,right+1));

i=left+1;

}

}

i++;

}

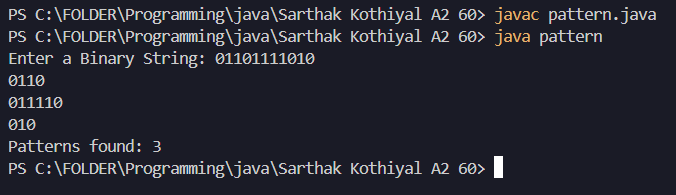
System.out.println("Patterns found: "+count);

sc.close();

}

}

**OUTPUT:**



**Practical No. 9**

Write a java program to create a class named 'Bank ' with the following data members:

Name of depositor

Address of depositor

Account Number

Balance in account

**Source Code:**

import java.util.Scanner;

public class bank {

private String name;

private String accnumber;

private String address;

private Double balance=0.00;

void setter(String n,String accnum,String add){

name=n;

accnumber=accnum;

address=add;

}

void showdetails(){

System.out.println("\nAccount Holder: " + name);

System.out.println("Account Number: " + accnumber);

System.out.println("Address: " + address);

System.out.println("Account Balance: " + balance); }

void deposit(double amm){

if(amm>0){

balance+=amm;

System.out.println("$" + amm + " deposited successfully...");

}

else{

System.out.println("Deposit amount cannot be negative...");

}

}

void withdraw(double amm){

if(amm<0)

System.out.println("Amount cannot be Negative...");

else if(amm>balance){

System.out.println("Amount cannot be more than Account Balance...");

}

else{

balance-=amm;

System.out.println(amm+" has been withdrawn...\nNew Balance: "+balance);

}

}

public static void main(String[] args) {

int acno=1000;

Scanner sc=new Scanner(System.in);

System.out.print("Enter number of Depositors: ");

int n=sc.nextInt();

bank bk[]=new bank[n];

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

bk[i]=new bank();

}

int count=0;

while(true){

int choice;

System.out.print("\n0. Create Account\n1. Show Details\n2. Deposit\n3. WithDraw\n4. Exit\nEnter your Choice: ");

choice=sc.nextInt();

sc.nextLine();

switch(choice){

case 0:{

if(count<n){

System.out.print("Enter Name: ");

String name=sc.nextLine();

System.out.print("Enter Address: ");

String address=sc.nextLine();

acno+=1;

String accNumber="ACC"+acno;

bk[count].setter(name,accNumber,address);

System.out.println("Account created successfully! Account Number: " + accNumber);

count++;

}

else{

System.out.println("Maximum number of accounts reached.");

}

break;

}

case 1:{

System.out.print("Enter Account Number: ");

String search=sc.nextLine();

boolean found=false;

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

if(bk[i].accnumber.equals(search)){

bk[i].showdetails();

found=true;

break;

}

}

if(!found){

System.out.println("Account not found!");

}

break;

}

case 2:{

System.out.print("Enter Account Number: ");

String search=sc.nextLine();

boolean found=false;

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

if(bk[i].accnumber.equals(search)){

System.out.print("Enter Amount to Deposit: ");

double val=sc.nextDouble();

bk[i].deposit(val);

found=true;

break;

}

}

if(!found){

System.out.println("Account not found!");

}

break;

}

case 3:{

System.out.print("Enter Account Number: ");

String search=sc.nextLine();

boolean found=false;

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

if(bk[i].accnumber.equals(search)) {

System.out.print("Enter Amount to Withdraw: ");

double val=sc.nextDouble();

bk[i].withdraw(val);

found=true;

break;

}

}

if(!found){

System.out.println("Account not found!");

}

break;

}

case 4:{

System.out.print("Program Ended...");

sc.close();

return;

}

default:{

System.out.println("Invlaid Choice...");

}

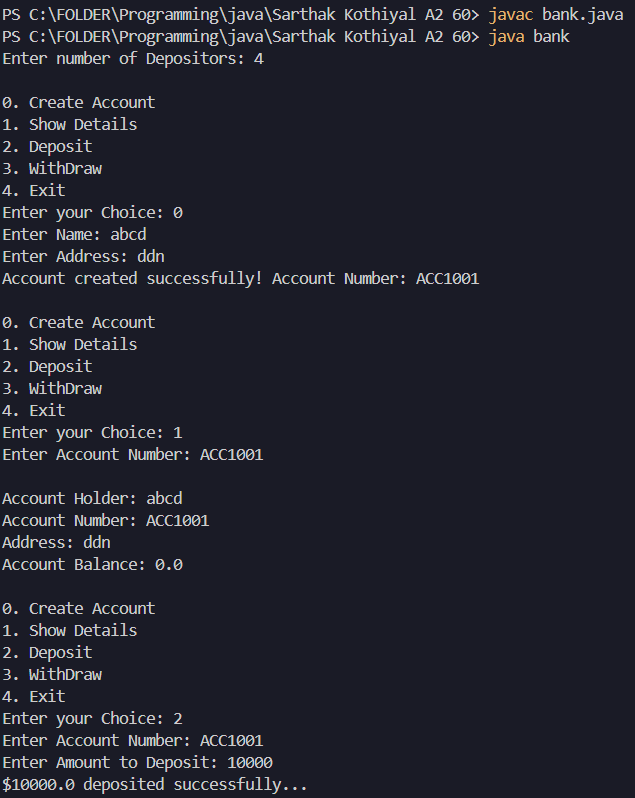
}

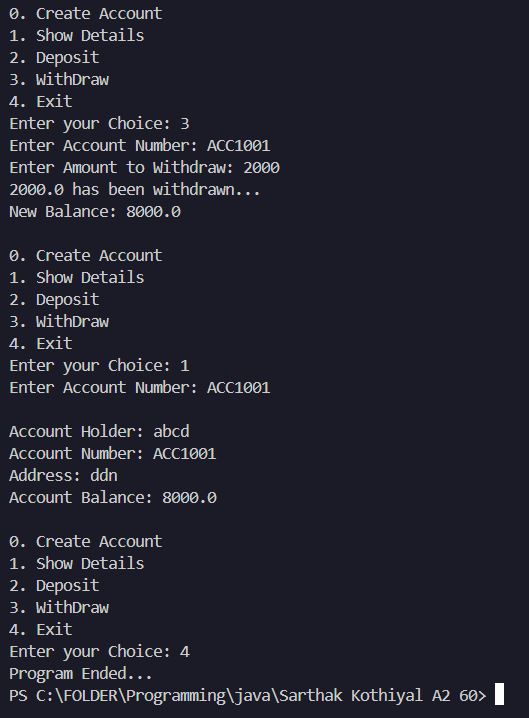
}

}

}

**OUTPUT:**





**PRACTICAL 10**

Define a class WordExample having the following description:

Data members/instance variables:

private String strdata : to store a sentence.

Parameterized Constructor

WordExample(String) : Accept a sentence which may be terminated by either’.’, ‘? ’or ’!’ only. The words may be separated by more than one blank space and are in UPPER CASE.

Member Methods: void countWord(): Find the number of words beginning and ending with a vowel.

void placeWord(): Place the words which begin and end with a vowel at the beginning, followed by the remaining words as they occur in the sentence.

**Source Code:**

import java.util.Scanner;

public class wordexample{

String strdata;

wordexample(String *s*){

this.strdata = s;

}

public void countWords() {

String[] words = strdata.split("\\s+");

int cnt = 0;

for (String word : words) {

if (!word.isEmpty() && checkVowel(word.charAt(0)) &&

checkVowel(word.charAt(word.length() - 1))) {

cnt++;

}

}

System.out.println("Numbers starting and ending with vowels: " + cnt);

}

public void placeWords() {

String[] words = strdata.split("\\s+");

StringBuilder vowelWords = new StringBuilder();

StringBuilder otherWords = new StringBuilder();

for (String word : words) {

if (!word.isEmpty() && checkVowel(word.charAt(0)) &&

checkVowel(word.charAt(word.length() - 1))) {

vowelWords.append(word + " ");

}

else{

otherWords.append(word + " ");

}

}

System.out.println("New Sentence is: " + vowelWords.toString() +

otherWords.toString());

}

public boolean checkVowel(char *c*) {

return "AEIOUaeiou".indexOf(c) != -1;

}

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

Scanner sc = new Scanner(System.in);

System.out.print("Enter a String: ");

String s = sc.nextLine();

if (s.endsWith(",") || s.endsWith("?") || s.endsWith("!")) {

wordexample obj = new wordexample(s);

obj.countWords();

obj.placeWords();

}

else {

System.out.println("Invalid String.");

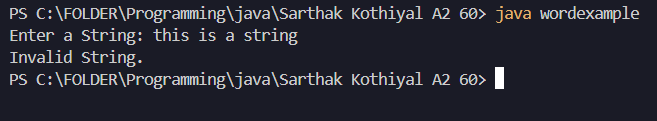
}

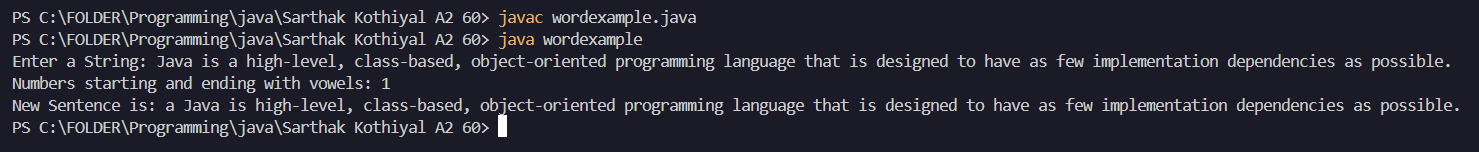
sc.close();

}

}

**OUTPUT:**





**PRACTICAL 11**

Write a Java program to create a class called ArrayDemo and overload arrayFunc() function.

void arrayFunc(int [], int) ➔To find all pairs of elements in an Array whose sum is equal to a given number.

**Source Code:**

import java.util.Arrays;

import java.util.Scanner;

public class arrayDemo {

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

public static void arraysFunction(int []a, int p, int []b, int q){

int []temp = new int[p+q];

for(int i=0; i<p+q; i++){

if(i<p){

temp[i] = a[i];

}

else{

temp[i] = b[i-p];

}

}

Arrays.sort(temp);

for(int i=0; i<p+q; i++){

if(i<p){

a[i] = temp[i];

}

else{

b[i-p] = temp[i];

}

}

}

public static void arraysFunction(int []arr, int k){

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

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

if(arr[i] + arr[j] == k){

System.out.println(arr[i]+" "+arr[j]);

}

}

}

}

public static void main(String[] args) {

System.out.print("Enter size of Array: ");

int n = sc.nextInt();

int []arr = new int[n];

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

arr[i] = sc.nextInt();

}

System.out.println("Enter target element: ");

int k =sc.nextInt();

arraysFunction(arr, k);

System.out.print("Enter p: ");

int p = sc.nextInt();

int []a = new int[p];

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

a[i] = sc.nextInt();

}

System.out.print("Enter q: ");

int q = sc.nextInt();

int []b = new int[q];

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

b[i] = sc.nextInt();

}

arraysFunction(a, p, b, q);

for(int i : a)

System.out.print(i+" ");

System.out.println();

for(int j : b)

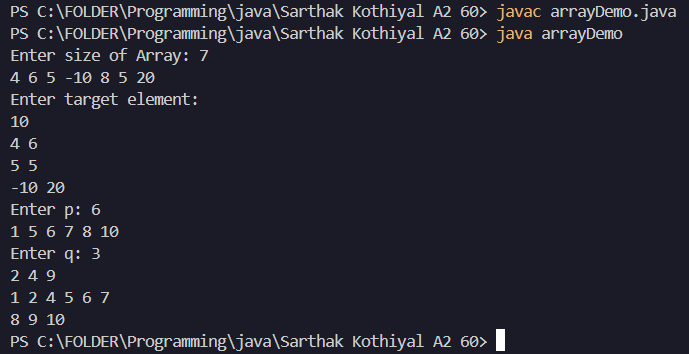
System.out.print(j+" ");

System.out.println();

}

}

**OUTPUT:**



**PRACTICAL 12**

Write a java program to calculate the area of a rectangle, a square and a circle. Create an abstract class 'Shape' with three abstract methods namely rectangleArea() taking two parameters, squareArea() and circleArea() taking one parameter each. Now create another class ‘Area’ containing all the three methods rectangleArea(),squareArea() and circleArea() for printing the area of rectangle, square and circle respectively. Create an object of class Area and call all the three methods.

**Source Code:**

import java.text.DecimalFormat;

import java.util.Scanner;

abstract class Shape {

abstract void rectangleArea(double length, double breadth);

abstract void squareArea(double side);

abstract void circleArea(double radius);

}

class area extends Shape {

void rectangleArea(double length, double breadth) {

System.out.println("Area of Rectangle: " + (length \* breadth));

}

void squareArea(double side) {

System.out.println("Area of Square: " + (side \* side));

}

void circleArea(double radius) {

DecimalFormat d=new DecimalFormat("#.##");

System.out.println("Area of Circle: " + d.format(Math.PI \* radius \* radius));

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

area obj = new area();

System.out.print("Enter length and breadth of rectangle: ");

double length = sc.nextDouble();

double breadth = sc.nextDouble();

obj.rectangleArea(length, breadth);

System.out.print("Enter side of square: ");

double side = sc.nextDouble();

obj.squareArea(side);

System.out.print("Enter radius of circle: ");

double radius = sc.nextDouble();

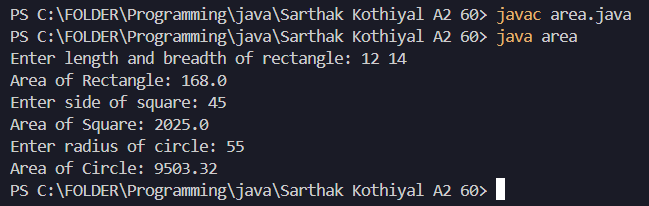
obj.circleArea(radius);

sc.close();

}

}

**OUTPUT:**



**PRACTICAL 13**

Write a java program to implement abstract class and abstract method with following details:

Create an abstract Base Class Temperature

Data members: double temp;

Method members: void setTempData(double), abstact void changeTemp()

Sub Class Fahrenheit (subclass of Temperature)

Data members: double ctemp;

method member: Override abstract method changeTemp() to convert Fahrenheit temperature into degree Celsius by using formula C=5/9\*(F-32) and display converted temperature

Sub Class Celsius (subclass of Temperature)

Data member: double ftemp;

Method member: Override abstract method changeTemp() to convert degree Celsius into fahrenheit temperature by using formula F=9/5\*c+32 and display converted temperature

**Source Code:**

import java.text.DecimalFormat;

import java.util.Scanner;

abstract class TemperatureBase {

double temp;

void setTempData(double temp) {

this.temp = temp;

}

abstract void changeTemp();

}

class Fahrenheit extends TemperatureBase {

double ctemp;

void changeTemp() {

ctemp = (5.0 / 9) \* (temp - 32);

System.out.println("Temperature in Celsius: " + temperature.d.format(ctemp));

}

}

class Celsius extends TemperatureBase {

double ftemp;

void changeTemp() {

ftemp = (9.0 / 5) \* temp + 32;

System.out.println("Temperature in Fahrenheit: " + temperature.d.format(ftemp));

}

}

public class temperature {

static DecimalFormat d=new DecimalFormat("#.##");

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

Fahrenheit f = new Fahrenheit();

System.out.print("Enter temperature in Fahrenheit: ");

double tempF = sc.nextDouble();

f.setTempData(tempF);

f.changeTemp();

Celsius c = new Celsius();

System.out.print("Enter temperature in Celsius: ");

double tempC = sc.nextDouble();

c.setTempData(tempC);

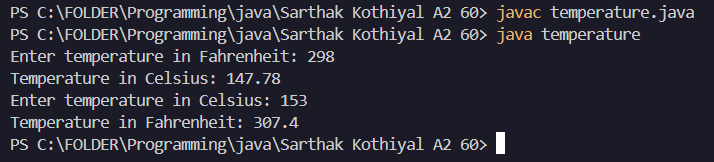
c.changeTemp();

sc.close();

}

}

**OUTPUT:**



**PRACTICAL 14**

Write a java program to create an interface that consists of a method to display volume () as an abstract method and redefine this method in the derived classes to suit their requirements.

Create classes called Cone, Hemisphere and Cylinder that implements the interface. Using these three classes, design a program that will accept dimensions of a cone, cylinder and hemisphere interactively and display the volumes.

**Source Code:**

import java.text.DecimalFormat;

import java.util.Scanner;

interface Shape {

void volume();

}

class Cone implements Shape {

double radius, height;

Cone(double r, double h) {

radius = r;

height = h;

}

public void volume() {

double vol = (1.0 / 3) \* Math.PI \* radius \* radius \* height;

System.out.println("Volume of Cone: " + volume.d.format(vol));

}

}

class Hemisphere implements Shape {

double radius;

Hemisphere(double r) {

radius = r;

}

public void volume() {

double vol = (2.0 / 3) \* Math.PI \* radius \* radius \* radius;

System.out.println("Volume of Hemisphere: " + volume.d.format(vol));

}

}

class Cylinder implements Shape {

double radius, height;

Cylinder(double r, double h) {

radius = r;

height = h;

}

public void volume() {

double vol = Math.PI \* radius \* radius \* height;

System.out.println("Volume of Cylinder: " + volume.d.format(vol));

}

}

public class volume {

static DecimalFormat d=new DecimalFormat("#.##");

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter radius and height of the cone: ");

double rCone = sc.nextDouble();

double hCone = sc.nextDouble();

Cone cone = new Cone(rCone, hCone);

System.out.print("Enter radius of the hemisphere: ");

double rHemisphere = sc.nextDouble();

Hemisphere hemisphere = new Hemisphere(rHemisphere);

System.out.print("Enter radius and height of the cylinder: ");

double rCylinder = sc.nextDouble();

double hCylinder = sc.nextDouble();

Cylinder cylinder = new Cylinder(rCylinder, hCylinder);

cone.volume();

hemisphere.volume();

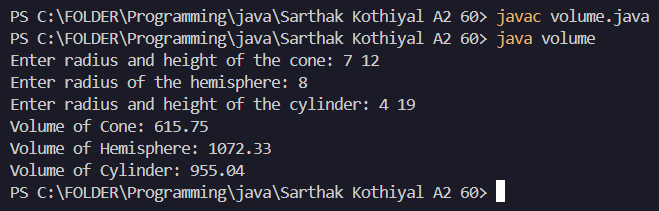
cylinder.volume();

sc.close();

}

}

**OUTPUT:**



**PRACTICAL 15**

Design a student management system using Java packages. The system should manage information about students, courses, and their enrolments. You will organize the functionality across multiple packages and use object-oriented principles such as encapsulation.

1. In the student package, create a Student class with fields for studentId, name, and age. Include a constructor to initialize these fields and a getStudentInfo() method to return the student's details.

2. In the course package, create a Course class with fields for courseId, courseName, and credits. Include a constructor and a getCourseInfo() method to return the course's details.

3. In the enrollment package, create an Enrollment class that holds references to Student and Course objects. Include a constructor and a getEnrollmentInfo() method to return the enrollment details.

4. In the default package, create instances of Student, Course, and Enrolment, then print details using the getStudentInfo(), getCourseInfo(), and getEnrollmentInfo() methods.

**Source Code:**

StudentManagementSystem (Main file):

import student.Student;

import course.Course;

import enrollment.Enrollment;

public class studentManagementSystem {

public static void main(String[] args) {

Student student = new Student(101, "ABC", 20);

Course course = new Course(408, "Java Programming", 4);

Enrollment enrollment = new Enrollment(student, course);

System.out.println(student.getStudentInfo());

System.out.println(course.getCourseInfo());

System.out.println(enrollment.getEnrollmentInfo());

}

}

Student.java

package student;

public class Student {

private int studentId;

private String name;

private int age;

public Student(int studentId, String name, int age) {

this.studentId = studentId;

this.name = name;

this.age = age;

}

public String getStudentInfo() {

return "Student ID: " + studentId + ", Name: " + name + ", Age: " + age;

}

}

Course.java

package course;

public class Course {

private int courseId;

private String courseName;

private int credits;

public Course(int courseId, String courseName, int credits) {

this.courseId = courseId;

this.courseName = courseName;

this.credits = credits;

}

public String getCourseInfo() {

return "Course ID: " + courseId + ", Course Name: " + courseName + ", Credits: " + credits;

}

}

Enrollment.java

package enrollment;

import student.Student;

import course.Course;

public class Enrollment {

private Student student;

private Course course;

public Enrollment(Student student, Course course) {

this.student = student;

this.course = course;

}

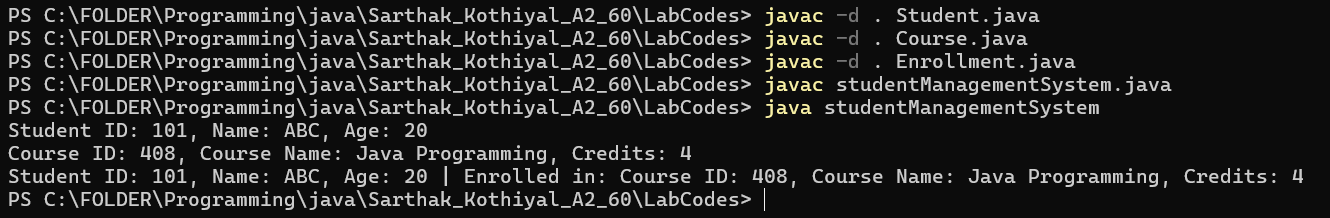
public String getEnrollmentInfo() {

return student.getStudentInfo() + " | Enrolled in: " + course.getCourseInfo();

}

}

**OUTPUT:**

****

**PRACTICAL 16**

Write a java program to accept and print the employee details during runtime. The details will include employee id, name, dept\_ Id. The program should raise an exception if user inputs incomplete or incorrect data. The entered value should meet the following conditions:

(i) First Letter of employee name should be in capital letter.

(ii) Employee id should be between 2001 and 5001

(iii) Department id should be an integer between 1 and 5.

If the above conditions are not met then the application should raise specific exception else should complete normal execution.

**Source Code:**

import java.util.\*;

class exp extends RuntimeException{

void showid(){

System.out.println("The entered ID details does not meet the correct format...");

}

void showname(){

System.out.println("The entered NAME details does not meet the correct format...");

}

void showdptid(){

System.out.println("The entered Department ID details does not meet the correct format...");

}

}

public class employeeException {

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

System.out.print("Enter Employee Name: ");

String name=sc.nextLine();

System.out.print("Enter Employee ID: ");

int id=sc.nextInt();

System.out.print("Enter Department ID: ");

int dpt\_id=sc.nextInt();

try{

if(name.charAt(0)<65 || name.charAt(0)>92){

exp obj=new exp();

throw obj;

}

}

catch(exp e){

e.showname();

}

try{

if(id<2001 || id>5001){

exp obj=new exp();

throw obj;

}

}

catch(exp e){

e.showid();

}

try{

if(dpt\_id<1 || dpt\_id>5){

exp obj=new exp();

throw obj;

}

}

catch(exp e){

e.showdptid();

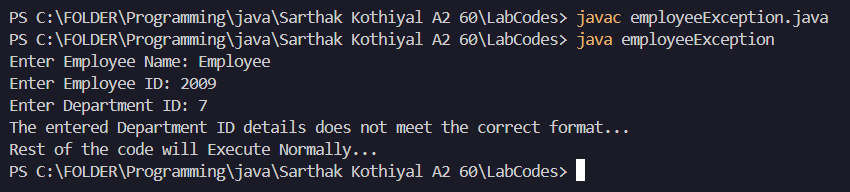
}

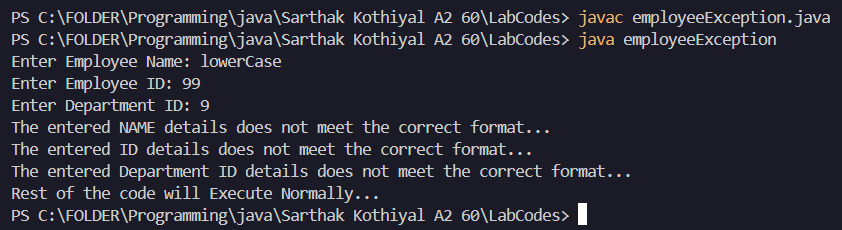
System.out.println("Rest of the code will Execute Normally...");

}

}

**OUTPUT:**

****

****

**PRACTICAL 17**

Create a class MyCalculator which consists of a single method power(int, int). This method takes two integers, n and p, as parameters and finds np .

If either n or p is negative, then the method must throw an exception which says "n and p should be non-negative".

**Source Code:**

import java.util.\*;

class exp extends RuntimeException{

exp(String str){

super(str);

}

}

public class MyCalculator {

static double power(int n,int p){

return Math.pow(n,p);

}

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

System.out.print("Enter value of n: ");

int n=sc.nextInt();

System.out.print("Enter value of p: ");

int p=sc.nextInt();

try{

if(n<0 || p<0){

exp obj=new exp("Value of 'n' or 'p' cannot be Negative...");

throw obj;

}

if(n==0 || p==0){

exp obj=new exp("Value of 'n' or 'p' cannot be Zero...");

throw obj;

}

System.out.println(power(n,p));

}

catch(exp e){

System.out.println(e.getMessage());

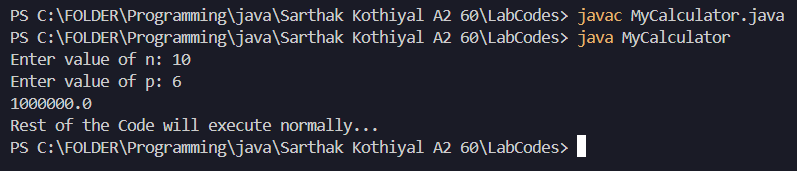
}

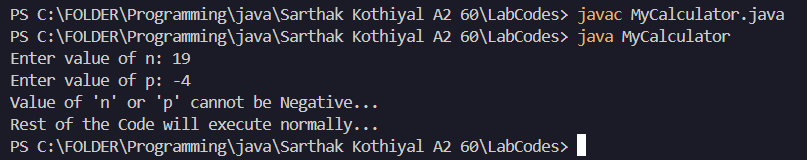
System.out.println("Rest of the Code will execute normally...");

}

}

**OUTPUT:**

****

****

**PRACTICAL 18**

Write a java file handling program to count and display the number of palindromes resent in a text file "myfile.txt".

**Source Code:**

import java.io.\*;

import java.util.Scanner;

class countpalindrome{

static boolean check(String str){

if(str==null || str.isEmpty())

return false;

String word=str.toLowerCase();

int i=0,j=word.length()-1;

while(i<j){

if(word.charAt(i)!=word.charAt(j)){

return false;

}

i++;j--;

}

return true;

}

public static void main(String[] args) throws Exception{

int count=0;

Scanner sc=new Scanner(new File("myfile.txt"));

while(sc.hasNext()){

String str=sc.next();

if(check(str)){

count++;

System.out.println(str);

}

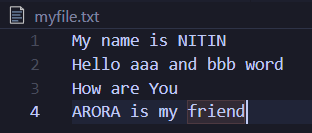
}

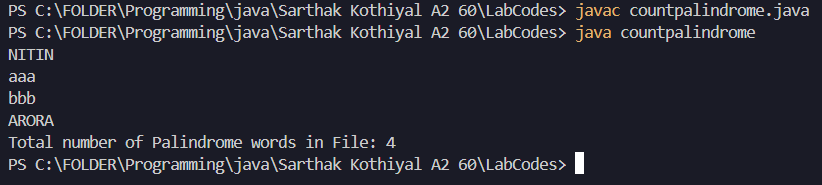
System.out.println("Total number of Palindrome words in File: "+count);

}

}

**OUTPUT:**

****

****

**PRACTICAL 19**

Write a program MultiThreads that creates two thread some thread with the name CSthread and the other thread named ITthread. Each thread should display its respective name and execute after a gap of 500 milliseconds. Each thread should also display a number indicating the number of times it got a chance to execute.

**Source Code:**

class MyThread extends Thread {

    String name;

    MyThread(String *name*) {

        this.name = name;

        setName(name);

    }

    public void run() {

        for (int i = 1; i <= 5; i++) {

            System.out.println(name + " executed " + i);

            try {

                Thread.sleep(500);

            }

            catch (InterruptedException *e*) {

                System.out.println("Thread " + name + " was interrupted.");

                return;

            }

        }

    }

}

public class WishMultithread {

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

        MyThread csThread = new MyThread("CSthread");

        MyThread itThread = new MyThread("ITthread");

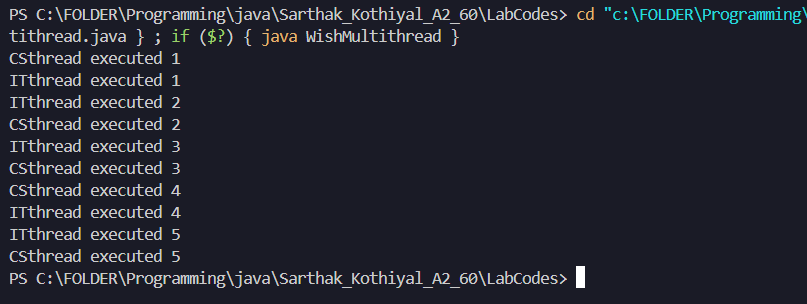
        csThread.start();

        itThread.start();

    }

}

**OUTPUT:**

****

**PRACTICAL 20**

Write a java program for to solve producer consumer problem in which a producer produce a value and consumer consume the value before producer generate the next value.

**Source Code:**

import java.util.\*;

class Pc1{

    LinkedList<Integer> list=new LinkedList<>();

    int capacity=5;

    public void produce() throws Exception{

        int value=0;

        while(true){

            synchronized(this){

                while(list.size()==capacity){

                    wait();

                }

                System.out.println("Producer is going to produce: "+value);

                list.add(value++);

                notify();

                Thread.sleep(500);

            }

        }

    }

    public void consume() throws Exception{

        while(true){

            synchronized(this){

                while(list.size()==0){

                    wait();

                }

                System.out.println("Consumer is going to Consume: "+list.removeFirst());

                notify();

                Thread.sleep(500);

            }

        }

    }

}

class A extends Thread{

    Pc1 p;

    public A(Pc1 *p*){

        this.p=p;

    }

    public void run(){

        try{

            p.produce();

        }

        catch(Exception *e*){

            System.out.println(e);

        }

    }

}

class B extends Thread{

    Pc1 p;

    public B(Pc1 *p*){

        this.p=p;

    }

    public void run(){

        try{

            p.consume();

        }

        catch(Exception *e*){

            System.out.println(e);

        }

    }

}

public class producerConsumer{

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

        Pc1 p=new Pc1();

        A t1=new A(p);

        B t2=new B(p);

        t1.start();

        t2.start();

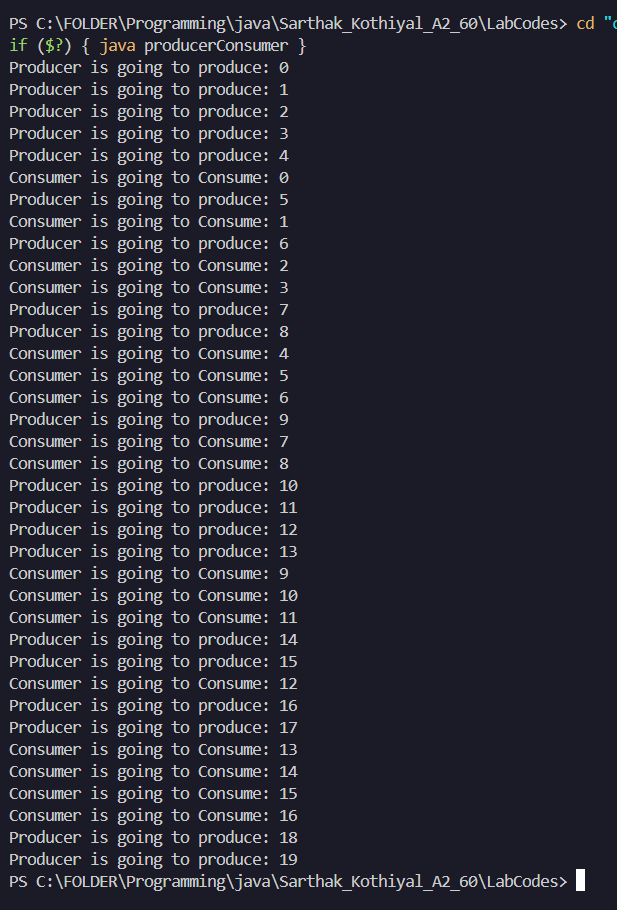
        t1.join();

        t2.join();

    }

}

**OUTPUT:**

****

**PRACTICAL 21**

Write a method removeEvenLength that takes an ArrayList of Strings as a parameter and that removes all of the strings of even length from the list. (Use ArrayList)

**Source Code:**

import java.util.\*;

public class RemoveEven {

    public void removeEvenLength(ArrayList<String> *list*) {

        Iterator<String> it = list.iterator();

        while (it.hasNext()) {

            String str = it.next();

            if (str.length() % 2 == 0) {

                it.remove();

            }

        }

    }

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

        Scanner sc=new Scanner(System.in);

        ArrayList<String> list = new ArrayList<String>();

        System.out.print("Enter number of Elements: ");

        int n=sc.nextInt();

        sc.nextLine();

        for(int i=0;i<n;i++){

            System.out.print("Enter element: ");

            String str=sc.nextLine();

            list.add(str);

        }

        RemoveEven obj = new RemoveEven();

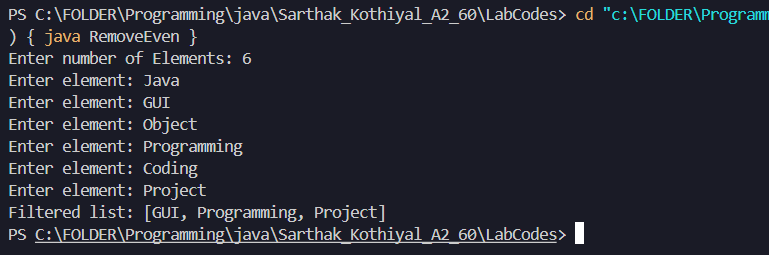
        obj.removeEvenLength(list);

        System.out.println("Filtered list: " + list);

    }

}

**OUTPUT:**



**PRACTICAL 22**

Write a method swapPairs that switches the order of values in an ArrayList of Strings in a pairwise fashion. Your method should switch the order of the first two values, then switch the order of the next two, switch the order of the next two, and so on.

**Source Code:**

import java.util.ArrayList;

import java.util.Scanner;

public class swappair {

public void swap(ArrayList<String> list){

for(int i=0;i<list.size()-1;i+=2){

String str=list.get(i);

list.set(i,list.get(i+1));

list.set(i+1,str);

}

}

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

ArrayList<String> list = new ArrayList<String>();

System.out.print("Enter number of Elements: ");

int n=sc.nextInt();

sc.nextLine();

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

System.out.print("Enter element: ");

String str=sc.nextLine();

list.add(str);

}

System.out.println("Original List: "+list);

swappair obj=new swappair();

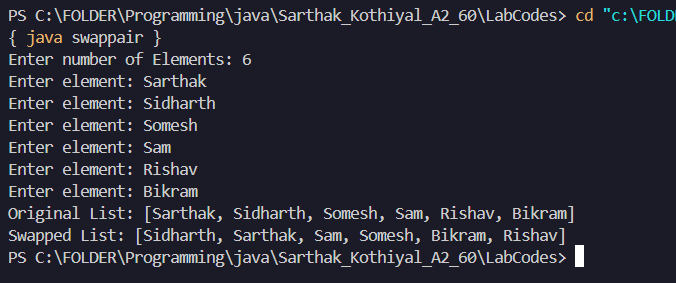
obj.swap(list);

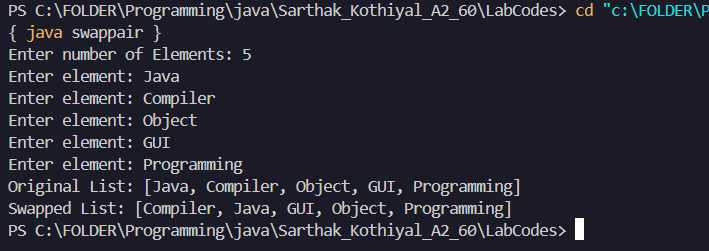
System.out.println("Swapped List: "+list);

}

}

**OUTPUT:**

****



**PRACTICAL 23**

Write a method called alternate that accepts two Lists of integers as its parameters and returns a new List containing alternating elements from the two lists, in the following order:

• First element from first list

• First element from second list

• Second element from first list

• Second element from second list

• Third element from first list

• Third element from second list

**Source Code:**

import java.util.\*;

public class duallist {

LinkedList<Integer> list3=new LinkedList<Integer>();

public void alternate(LinkedList<Integer> list1,LinkedList<Integer> list2){

int m=list1.size(),i=0;

int n=list2.size(),j=0;

while(i<m && j<n){

list3.addLast(list1.get(i++));

list3.addLast(list2.get(j++));

}

while(i<m){

list3.addLast(list1.get(i++));

}

while(j<n){

list3.addLast(list2.get(j++));

}

}

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

int m=0;

LinkedList<Integer> list1=new LinkedList<Integer>();

System.out.print("Enter number of Elements in list1: ");

m=sc.nextInt();

System.out.print("Enter Elements in list1: ");

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

list1.addLast(sc.nextInt());

}

int n=0;

LinkedList<Integer> list2=new LinkedList<Integer>();

System.out.print("Enter number of Elements in list2: ");

n=sc.nextInt();

System.out.print("Enter Elements in list2: ");

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

list2.addLast(sc.nextInt());

}

duallist obj=new duallist();

obj.alternate(list1, list2);

System.out.println("List 1: "+list1);

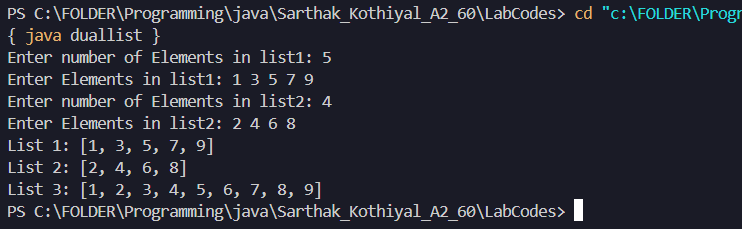
System.out.println("List 2: "+list2);

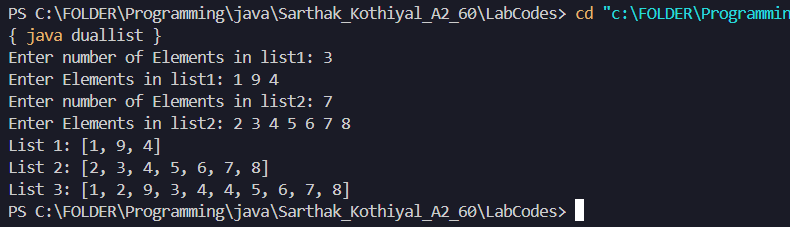
System.out.println("List 3: "+obj.list3);

}

}

**OUTPUT:**





**PRACTICAL 24**

Write a GUI program to develop an application that receives a string in one text field, and count number of vowels in a string and returns it in another text field, when the button named “CountVowel” is clicked. When the button named “Reset” is clicked it will reset the value of text field one and Text field two. When the button named “Exit” is clicked it will close the application.

**Source Code:**

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

class countvowel extends JFrame implements ActionListener{

JTextField tf1,tf2;

int count=0;

public countvowel(){

JLabel l1=new JLabel("Enter String: ");

tf1=new JTextField(20);

JButton jb1=new JButton("COUNT VOWEL");

JButton jb2=new JButton("RESET");

JButton jb3=new JButton("EXIT");

JLabel l2=new JLabel("Result: ");

tf2=new JTextField(20);

setLayout(new FlowLayout(FlowLayout.CENTER,30,10));

add(l1);

add(tf1);

add(jb1);

add(jb2);

add(jb3);

add(l2);

add(tf2);

jb1.addActionListener(this);

jb2.addActionListener(this);

jb3.addActionListener(this);

}

public void actionPerformed(ActionEvent e){

String s=e.getActionCommand();

String str=tf1.getText();

if(s.equals("EXIT")){

System.exit(0);

}

else if (s.equals("RESET")) {

count=0;

tf1.setText("");

tf2.setText("");

}

else{

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

char ch=str.charAt(i);

ch=Character.toLowerCase(ch);

if(ch=='a' || ch=='e' || ch=='i' || ch=='o' || ch=='u'){

count++;

}

}

tf2.setText(String.valueOf(count));

}

}

public static void main(String args[]) {

countvowel d = new countvowel();

d.setSize(400, 300);

d.setVisible(true);

}

}

**OUTPUT:**

