**Q-1**. **Write a program to take input as a command line argument. The information should be student name, course, university roll number and semester. Finally display this info.**

**Sol:**

public class LabQ1{

public static void main(String args[]){

String name;

String unvRno;

String course;

int sem;

name = args[0];

unvRno = args[1];

course = args[2];

sem = Integer.parseInt(args[3]);

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

System.out.println("University Roll No: "+unvRno);

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

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

}

}

**Output:**

Text

Description automatically generated

**Q-2**. **Using the Switch statement write a menu driven program to calculate the maturity amount of a bank deposit. The use must have the following options.**

**Option 1: Term Deposit Option 2: Recurring Deposit**

**For Option 1-> Accept principle(P), then rate of interest(R) and time period in years(N). Calculate and output the maturity amount(A) with the formula A=P[(1+R/100) ^N].**

**For Option 2-> Accept monthly instalments(P), rate of interest(R) and time in month(N). Calculate and output the maturity amount(A) receivable using the formula (P x N)\*[(N+1)/2]\*(R/100)\*(1/12).**

**If the option is incorrect then display an appropriate error message.**

**(Use scanner class for testing all inputs)**

**Sol:**

import java.util.Scanner;

public class LabQ2{

public static void main(String[] args) {

System.out.println("Option 1: TERM DEPOSIT");

System.out.println("Option 2: RECURRING DEPOSIT");

Scanner sc = new Scanner(System.in);

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

int option = sc.nextInt();

float P,R,A;

int N;

switch (option){

case 1:

System.out.print("Enter the Principle Amount: ");

P = sc.nextFloat();

System.out.print("Enter the Rate of Interest: ");

R = sc.nextFloat();

System.out.print("Enter the Time Period: ");

N = sc.nextInt();

float temp = (1+(R/100));

float res = 1;

while(N>0){

res \*= temp;

N--;

}

A = P\*res;

System.out.println(“Maturity Amount: Rs.”+A);

break;

case 2:

System.out.print("Enter the Principle Amount: ");

P = sc.nextFloat();

System.out.print("Enter the Rate of Interest: ");

R = sc.nextFloat();

System.out.print("Enter the Time in months: ");

N = sc.nextInt();

float interest = ((P\*N)\*(N+1)\*R)/2400;

A = (P\*N)+interest; ***// amount = Total deposits(monthly deposits \* number of months) + Interest***

System.out.println(“Maturity Amount: Rs.”+A);

break;

default:

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

}

}

}

**Output:**

Text

Description automatically generated

**Q-3. Write a program to find whether the given number are friendly pair or not**

**Sol:**

import java.util.Scanner;

public class LabQ3{

static int factors(int num){

int sum=0;

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

if(num%i==0){

sum+=i;

}

}

return sum;

}

public static void main(String[] args){

Scanner sc = new Scanner(System.in);

System.out.print("Enter number 1: ");

short num1 = sc.nextShort();

System.out.print("Enter number 2: ");

short num2 = sc.nextShort();

int sum1 = factors(num1);

int sum2 = factors(num2);

if(sum1/num1 == sum2/num2){

System.out.println(num1+" and "+num2+" are friendly pairs.");

}

else{

System.out.println("Not friendly pair.");

}

}

}

**Output:**

Text

Description automatically generated

**Q-4. Write to program to replace all 0s with 1 in an integer value.**

**Sol:**

import java.util.Scanner;

class LabQ4 {

public static void main(String[] args) {

int num;

int reverse = 0;

int remainder;

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

Scanner sc = new Scanner(System.in);

num = sc.nextInt();

int new\_num = 0;

while (num != 0) {

remainder = num % 10;

if (remainder == 0) {

remainder = 1;

}

reverse = reverse \* 10 + remainder;

num = num / 10;

}

while (reverse != 0) {

remainder = reverse % 10;

new\_num = new\_num \* 10 + remainder;

reverse = reverse / 10;

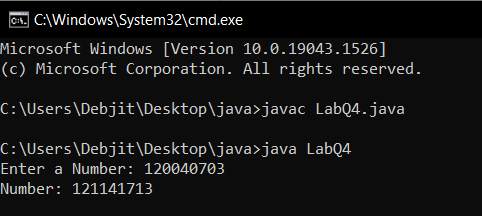
}

System.out.println("Number: " + new\_num);

}

}

**Output:**



**Q-5. Printing an array into a zig-zag fashion. Suppose you were given a array of integers and in zig-zag pattern. In zig-zag the 1st integer < 2nd integer > 3rd integer < 4th integer and so on.**

**Sol:**

import java.util.Scanner;

public class LabQ5 {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int size=sc.nextInt();

int[] arr = new int[size];

boolean x = true;

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

System.out.print("Enter element "+(i+1)+": ");

arr[i]=sc.nextInt();

}

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

if(x){

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

int temp=arr[i];

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

arr[i+1]=temp;

}

}

else{

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

int temp=arr[i];

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

arr[i+1]=temp;

}

}

x=!x;

}

System.out.print("Array in zig-zag fashion:- ");

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

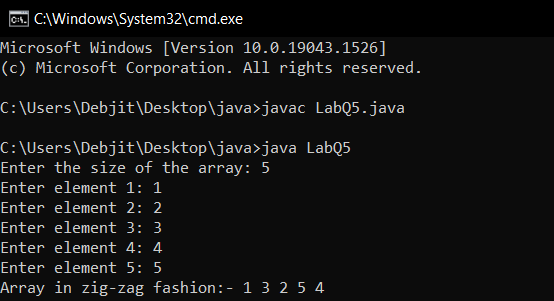
System.out.print(arr[i]+" ");

}

}

}

**Output:**



**Q-6. The problem to rearrange the positive and negative number. All the negative in the left side and positive in right side.**

**Sol:**

import java.util.Scanner;

public class LabQ6 {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int size = sc.nextInt();

int arr[] = new int[size];

int x = 0;

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

System.out.print("Enter element "+(i+1)+": ");

arr[i] = sc.nextInt();

if (arr[i] < 0)

x++;

}

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

for (int j = i + 1; j < size; j++) {

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

int temp = arr[j];

arr[j] = arr[i];

arr[i] = temp;

}

}

}

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

for (int j = i + 1; j < x; j++) {

if ((-1) \* arr[j] < (-1) \* arr[i]) {

int temp = arr[j];

arr[j] = arr[i];

arr[i] = temp;

}

}

}

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

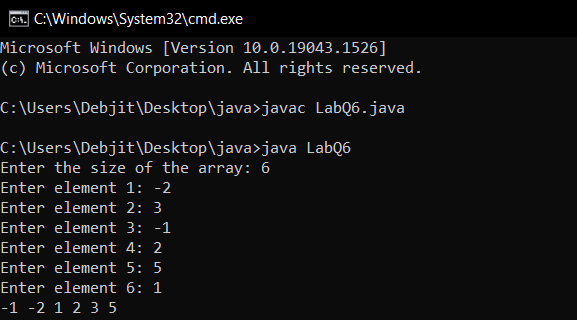
System.out.print(arr[i] + " ");

}

}

}

**Output:**



**Q-7. Write a program to find saddle point co-ordinates in given matrix. A saddle point is an element of the matrix. Which is minimum in its row and maximum of its column.**

**Sol:**

import java.util.Scanner;

public class LabQ7 {

static void saddlePoint(int[][] arr) {

int flag = 0;

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

int rowMin = arr[i][0];

int col = 0;

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

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

rowMin = arr[i][j];

col = j;

}

}

int check = 1;

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

if (arr[j][col] > rowMin) {

check = 0;

break;

}

}

if (check == 1) {

flag = 1;

System.out.println("Saddle Point is : " + rowMin);

}

}

if (flag != 1) {

System.out.println("Saddle not found");

}

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int row = sc.nextInt();

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

int col = sc.nextInt();

int[][] arr = new int[row][col];

System.out.println("Enter the element:-");

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

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

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

}

}

sc.close();

saddlePoint(arr);

}

}

**Output:**

Text

Description automatically generated

**Q-8. Write a program to find all patterns in a string. The given string will have 0’s and 1’s. Find total number of this pattern – 0(1+)0 occurring in strings and output the count od occurrence of string.**

**Sol:**

import java.util.Scanner;

public class LabQ8 {

public static void main(String[] arg) {

Scanner sc = new Scanner(System.in);

String s = sc.next();

sc.close();

int i = 0;

int j = 0;

int count = 0;

while (i < s.length() - 1) {

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

j = i + 1;

if (s.charAt(j) == '0') {

i++;

}

else {

while (j < s.length() && s.charAt(j) != '0') {

j++;

}

count++;

i = j;

}

}

else {

i++;

}

}

if (s.charAt(s.length() - 1) == '1')

count--;

System.out.println("Total 0(+1)0 pattern : " + count);

}

}

**Output:**

Text

Description automatically generated

**Q-9. Write a program in java to delete all vowels of a string using string buffer class.**

**Sol:**

import java.util.Scanner;

public class LabQ9 {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter the string: ");

String s = sc.nextLine();

sc.close();

StringBuffer str = new StringBuffer();

str.append(s.toLowerCase());

char[] vowel = {'a','e','i','o','u'};

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

for (char character : vowel) {

if(character==str.charAt(i)){

str.deleteCharAt(i);

}

}

}

System.out.println(str);

}

}

**Output:**

Text

Description automatically generated

**Q-10. Write a java program to create a class named 'Bank ' with the following**

**Data members:  
 1. Name of depositor  
 2. Address of depositor  
 3. Account Number  
 4. Balance in account**

### Class 'Bank' has a method for each of the following: 1 - Generate a unique account number for each depositor For first depositor, account number will be 1001, for second depositor it will be 1002 and so on 2 - Display information and balance of depositor 3 - Deposit more amount in balance of any depositor 4 - Withdraw some amount from balance deposited 5 - Change address of depositor

**After creating the class, do the following operations  
1 - Enter the information (name, address, account number, balance) of the depositors. Number of depositors is to be entered by user.  
2 - Print the information of any depositor.  
3 - Add some amount to the account of any depositor and then display final information of that depositor  
4 - Remove some amount from the account of any depositor and then display final information of that depositor  
5 - Change the address of any depositor and then display the final information of that depositor  
6 - Randomly repeat these processes for some other bank accounts.**

**Sol:**

import java.util.Random;

import java.util.Scanner;

public class Bank{

public String depositor;

public int accNumber;

public String address;

public float balance;

public void values(String depositor, float balance, String address){

this.depositor = depositor;

this.balance = balance;

this.address = address;

//Unique Account Number

Random rand = new Random();

this.accNumber = rand.nextInt(1000000);

}

public void deposit(float deposit){

balance+=deposit;

System.out.println("Amount deposited successfully...");

}

public void withdraw(float amount)

{

if(balance>amount){

balance-=amount;

System.out.println("Amount withdrawn successfully...");

}

else{

System.out.println("Insufficient balance !");

}

}

public void changeAddress(String address){

this.address = address;

}

public void display(){

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

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

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

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

}

public static void main(String[] args) {

Bank account = new Bank();

Scanner sc = new Scanner(System.in);

do{

System.out.println("1. Open account");

System.out.println("2. Deposit amount");

System.out.println("3. Withdraw amount");

System.out.println("4. Display Account details");

System.out.println("5. Change Address");

System.out.println("6. Exit\n");

System.out.print("\nEnter your choice: ");

int ch = sc.nextInt();

if(ch==1){

System.out.print("Enter your name: ");

String depositor = sc.nextLine();

System.out.print("Enter opening balance: ");

float balance = sc.nextFloat();

System.out.print("Enter your address: ");

String address = sc.nextLine();

account.values(depositor, balance, address);

}

else if(ch==2){

System.out.print("Enter amount to deposit: ");

account.deposit(sc.nextFloat());

}

else if(ch==3){

System.out.print("Enter amount to withdraw: ");

account.withdraw(sc.nextFloat());

}

else if(ch==4){

account.display();

}

else if(ch==5){

System.out.print("Enter new address: ");

account.changeAddress(sc.nextLine());

}

else{

break;

}

} while (true);

}

}

**Output:**

A picture containing text

Description automatically generated

**Q-11. Define a class wordExample having following description:-**

**The data members or instance variables are as below:**

**1- private String strData (stores sentence)**

**2- wordExample (pass string)**

**🡪Accept a sentence which may be terminated by either '.' or ‘?’ or '!' only)**

**🡪The words may be separated by more than one blank space, and all are in upper case.**

**Member methods:**

**1- Void countWord (find number of words beginning and ending with vowel)**

**2- Void placeWord (will place words which are beginning and ending with a vowel at the beginning followed by remaining words as they occur in sentence)**

**Sol:**

import java.util.Scanner;

public class WordExample {

private String strData;

boolean flag;

WordExample(String strData){

int len = strData.length();

char lastChar = strData.charAt(len-1);

if(lastChar=='?' || lastChar=='.' || lastChar=='!'){

this.strData=strData.toUpperCase();

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

}

else{

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

flag = true;

}

}

public void countWord(){

int count = 0;

String[] words = this.strData.split(" ");

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

String currentWord=words[i];

int len=currentWord.length();

char start=currentWord.charAt(0);

char end=currentWord.charAt(len-1);

if(start=='A'||start=='E'||start=='I'||start=='O'||start=='U'){

if(end=='A'||end=='E'||end=='I'||end=='O'||end=='U'){

count++;

}

}

}

System.out.println("Number of words beginning and ending with vowel: "+count);

}

public void placeWord(){

String withVowel = "";

String withoutVowel = "";

String[] words = this.strData.split(" ");

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

String currentWord=words[i];

int len=currentWord.length();

char start=currentWord.charAt(0);

char end=currentWord.charAt(len-1);

if(start=='A'||start=='E'||start=='I'||start=='O'||start=='U'){

if(end=='A'||end=='E'||end=='I'||end=='O'||end=='U'){

withVowel+=currentWord+" ";

}

else{

withoutVowel+=currentWord+" ";

}

}

else{

withoutVowel+=currentWord+" ";

}

}

System.out.println(withVowel+withoutVowel);

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

String sentence = sc.nextLine();

sc.close();

WordExample str = new WordExample(sentence);

if(str.flag == false){

str.countWord();

str.placeWord();

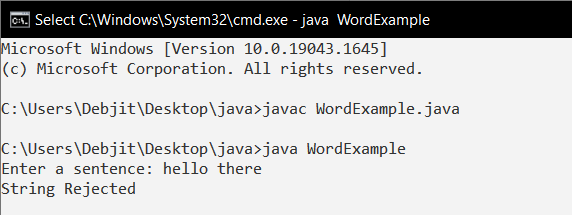
}

}

}

**Output:**

**Case 1-** String Rejected



**Case 2-** String Accepted

A screenshot of a computer

Description automatically generated with medium confidence

**Q-12. WAP 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.

Array numbers= [4, 6, 5, -10, 8, 5, 20], target=10

**Output :**

Pairs of elements whose sum is 10 are :  
4 + 6 = 10  
5 + 5 = 10  
-10 + 20 = 10

**🡪 void arrayFunc(int A[], int p, int B[], int q) –** Given two sorted arrays A and B of size p and q, Overload method arrayFunc() to merge elements of A with B by maintaining the sorted order i.e. fill A with first p smallest elements and fill B with remaining elements.

**Input :**  
int[] A = { 1, 5, 6, 7, 8, 10 }  
int[] B = { 2, 4, 9 }

**Output:**  
Sorted Arrays:  
A: [1, 2, 4, 5, 6, 7]  
B: [8, 9, 10]

**Sol:**

import java.util.Arrays;

public class ArrayDemo {

void arrayFunc(int[] arr, int num){

int count = 0;

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

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

int sum = arr[i]+arr[j];

if(sum == num)

{count++;}

}

}

System.out.println("Count: "+count);

}

void arrayFunc(int[] A, int p, int[] B, int q){

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

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

temp[i] = A[i];

}

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

temp[index] = B[i];

}

Arrays.sort(temp);

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

A[i] = temp[i];

}

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

B[i] = temp[index];

}

System.out.println("Elements of Array 1: ");

for (int i : A) {

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

}

System.out.println("\nElements of Array 2: ");

for (int i : B) {

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

}

}

public static void main(String[] args) {

ArrayDemo obj = new ArrayDemo();

**//function 1**

int[] arr = {4, 6, 5, -10, 8, 5, 20};

obj.arrayFunc(arr,10);

**//function 2**

int[] A = {1, 5, 6, 7, 8, 10 };

int[] B = {8, 9, 10};

obj.arrayFunc(A, A.length, B, B.length);

}

}

**Output:**

Text

Description automatically generated

**Q-13. 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.**

**Sol:**

public interface Area {

public void area(int r, int h);

public void area(int r);

}

public class LabQ13 {

public class Cone implements Area{

public void area(int r, int h){

float a = (1/3)\*3.14\*(r\*r)\*h;

System.out.println("Cone: "+a);

}

}

public class Hemisphere implements Area{

public void area(int r){

float a = (2/3)\*3.14\*(r\*r\*r);

System.out.println("Hemisphere: "+a);

}

}

public class Cylinder implements Area{

public void area(int r, int h){

float a = 3.14\*(r\*r)\*h;

System.out.println("Cylinder: "+a);

}

}

public static void main(String[] args) {

Cone cone = new Cone();

Hemisphere hemisphere = new Hemisphere();

Cylinder cylinder = new Cylinder();

cone.area(3,5);

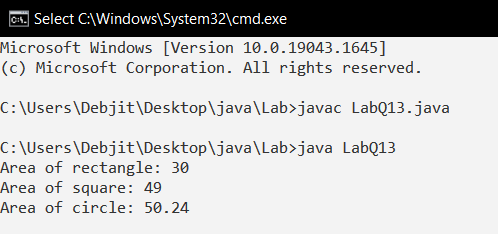
hemisphere.area(3);

cylinder.area(3,5);

}

}

**Output:**



**Q-14. Write a java program to implement abstract class and abstract method with following details: -**

**Create a 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

**Sol:**

abstract class Temperature {

float temp;

void setTempData(float a) {

temp = a;

}

abstract void changeTemp();

}

class Fahrenheit extends Temperature {

float ctemp;

public void changeTemp() {

ctemp = ((temp - 32) \* 5) / 9;

System.out.println("Fahrenheit: " + ctemp);

}

}

class Celsius extends Temperature {

double ftemp;

public void changeTemp() {

ftemp = (9 / 5 \* (temp + 32));

System.out.println("Celsius: " + ftemp);

}

}

public class LabQ14 {

public static void main(String[] args) {

Fahrenheit obj = new Fahrenheit();

obj.setTempData(95);

obj.changeTemp();

Celsius obj1 = new Celsius();

obj1.setTempData(95);

obj1.changeTemp();

}

}

**Output:**

Text

Description automatically generated

**Q-15. 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.**

**Volume of cone = (1/3)πr2h  
Volume of hemisphere = (2/3)πr3   
Volume of cylinder = πr2h**

**Sol:**

import java.util.Scanner;

interface Vol {

public void volume();

}

class cone implements Vol {

int r, h;

double pi = 3.14;

cone(int r, int h) {

this.r = r;

this.h = h;

}

public void volume() {

System.out.println("Volume of cone: " + (pi \* r \* r \* h) / 3);

}

}

class hemisphere implements Vol {

int r;

double pi = 3.14;

hemisphere(int r) {

this.r = r;

}

public void volume() {

System.out.println("Volume of hemisphere: " + (2 \* pi \* r \* r \* r) / 3);

}

}

class cylinder implements Vol{

int m, n;

double pi = 3.14;

cylinder(int m, int n) {

this.m = m;

this.n = n;

}

public void volume() {

System.out.println("Volume of cyliner: " + pi \* m \* m \* n);

}

}

public class LabQ15 {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Radius: ");

int r = sc.nextInt();

System.out.print("Height: ");

int h = sc.nextInt();

cone obj = new cone(r, h);

obj.volume();

cylinder obj1 = new cylinder(r, h);

obj1.volume();

hemisphere obj2 = new hemisphere(r);

obj2.volume();

}

}

**Output:**

Text, letter

Description automatically generated

**Q-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.**

**Sol:**

import java.util.Scanner;

class NewException extends Exception {

NewException(String s) {

super(s);

}

}

class Employee {

String name;

int empid, dept\_id;

void getdata(int empid, String name, int dept\_id) {

this.empid = empid;

this.name = name;

this.dept\_id = dept\_id;

}

boolean checkname() {

char x = name.charAt(0);

return Character.isUpperCase(x);

}

boolean checkempid() {

return empid > 2001 && empid < 5001;

}

boolean checkdept() {

return dept\_id > 1 && dept\_id < 5;

}

void printdata() {

System.out.println("Name: " + name + "\nEmployee id: " + empid + "\nDepartment id: " +

dept\_id);

}

}

public class LabQ16 {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int empid = sc.nextInt();

sc.nextLine();

System.out.print("\nEnter name: ");

String name = sc.nextLine();

System.out.print("\nEnter department id: ");

int dept\_id = sc.nextInt();

Employee obj = new Employee();

try {

obj.getdata(empid, name, dept\_id);

if (obj.checkname() && obj.checkempid() && obj.checkdept()) {

obj.printdata();

} else

throw new NewException("Wrong or incoplete data input");

} catch (NewException e) {

System.out.println("Exception: " + e.getMessage());

}

}

}

**Output:**

**Case 1-**

Graphical user interface, text, application, email

Description automatically generated

**Case 2-** Exception Raised

Graphical user interface, text, application

Description automatically generated

**Q-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".**

**Input Format**

Each line of the input contains two integers, n and p .

**Output Format**

Each line of the output contains the result ,if neither of n and p is negative.

Otherwise the output contains "n and p should be non-negative".

**Sample Input**

3 5

2 4

0 0

-1 -2

-1 3

**Sample Output**

243

16

java.lang.Exception: n and p should not be zero.

java.lang.Exception: n or p should not be negative.

java.lang.Exception: n or p should not be negative.

**Explanation:**

In the first two cases, both **n** and **p** are positive. So, the power function returns the answer correctly.

In the third case, both **n** and **p** are zero. So, the exception, "**n** and **p** should not be zero.” is printed.

In the last two cases, at least one out of **n** and **p** is negative. So, the exception, "**n** or **p** should not be negative.” is printed for these two cases.

**Sol:**

import java.util.Scanner;

class MyCalculator {

public long power(int n, int p) throws Exception {

if (n == 0 && p == 0)

throw new Exception("n and p should not be zero.");

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

throw new Exception("n or p should not be negative.");

else

return (long) (Math.pow(n, p));

}

}

public class LabQ17 {

public static final MyCalculator my\_calculator = new MyCalculator();

public static final Scanner in = new Scanner(System.in);

public static void main(String[] args) {

while (in.hasNextInt()) {

int n = in.nextInt();

int p = in.nextInt();

try {

System.out.println(my\_calculator.power(n, p));

} catch (Exception e) {

System.out.println(e);

}

}

}

}

**Output:**

Graphical user interface, text, application, email

Description automatically generated

**Q-18.** **Write a java file handling program to count and display the number of palindromes present in a text file "myfile.txt".**

***Example:*** If the file "myfile.txt" contains the following lines,

My name is NITIN

Hello aaa and bbb word

How are You

ARORA is my friend

**Output will be => 4 // NITIN, aaa, bbb, ARORA**

**Sol:**

import java.io.\*;

import java.util.\*;

public class LabQ18 {

static int ispalindrome(String word) {

String rev = "";

char ch;

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

ch = word.charAt(i);

rev = ch + rev;

}

if (rev.equals(word))

return 1;

else

return 0;

}

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

File f = new File("MyFile.txt");

FileReader fr = new FileReader(f);

BufferedReader br = new BufferedReader(fr);

String str;

String[] words = null;

int count = 0;

while ((str = br.readLine()) != null) {

words = str.split(" ");

for (String word : words) {

int res = ispalindrome(word);

if (res == 1) {

count++;

System.out.println(word);

}

}

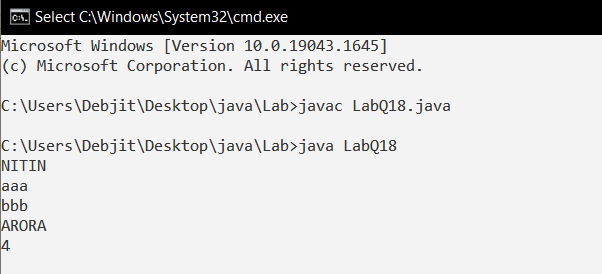
}

System.out.println(count);

}

}

**Output:**



**Q-19. Write down a program multithreads that creates two threads with the name CSthread and ITthread. Each thread should display its respective name and execute after a gap of 500ms { sleep(500) }. Each thread should also display a number indicating the number of times it got a chance to execute.**

**Sol:**

class CSthread extends Thread {

String t1;

public CSthread(String name) {

this.t1 = name;

}

public void run() {

System.out.println("Thread name "+t1);

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

try {

Thread.sleep(500);

} catch (InterruptedException e) {

System.out.println(e);

}

System.out.println("CSthread "+i);

}

System.out.println("Exiting CSthread");

}

}

class ITthread extends Thread{

String t1;

public ITthread(String name) {

this.t1 = name;

}

public void run() {

System.out.println("Thread name "+t1);

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

try {

Thread.sleep(500);

} catch (InterruptedException e) {

System.out.println(e);

}

System.out.println("ITthread " +i);

}

System.out.println("Exiting ITthread");

}

}

public class LabQ19 {

public static void main(String[] args) {

CSthread t1=new CSthread("CSthread");

ITthread t2=new ITthread("ITthread");

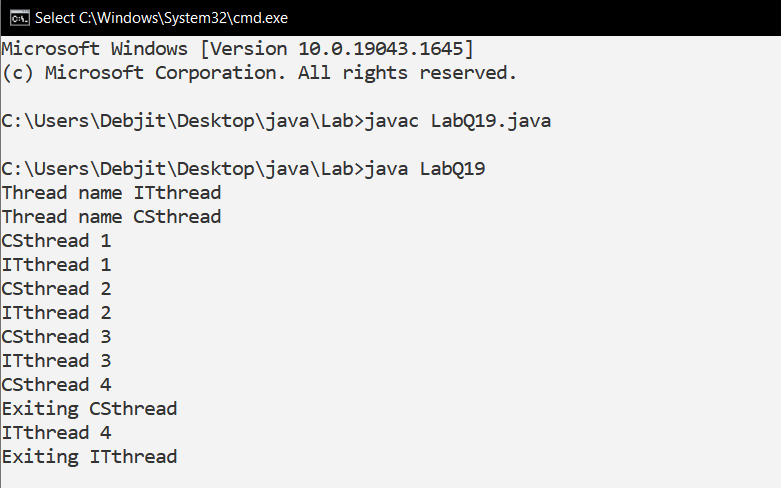
t1.start();

t2.start();

}

}

**Output:**



**Q-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.**

**Sol:**

class Company {

int n;

boolean t = true;

synchronized void produce\_item(int n) {

if (!t) {

try {

wait();

} catch (Exception e) {

}

}

this.n = n;

System.out.println("Produced : " + this.n);

t = false;

notify();

}

synchronized int consume\_item() {

if (t) {

try {

wait();

} catch (Exception e) {

}

}

System.out.println("Consumed : " + this.n);

t = true;

notify();

return this.n;

}

}

class Producer extends Thread {

Company c;

int count = 1;

Producer(Company c) {

this.c = c;

}

public void run() {

while (count < 10) {

this.c.produce\_item(count);

try {

Thread.sleep(5000);

} catch (Exception e) {

}

count++;

if(count == 9)

System.exit(0);

}

}

}

class Consumer extends Thread {

Company c;

Consumer(Company c) {

this.c = c;

}

public void run() {

while (true) {

this.c.consume\_item();

try {

Thread.sleep(1000);

} catch (Exception e) {

}

}

}

}

public class LabQ20 {

public static void main(String[] args) {

Company comp = new Company();

Producer p = new Producer(comp);

Consumer c = new Consumer(comp);

p.start();

c.start();

}

}

**Output:**

Text

Description automatically generated

**Q-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)**

**Sol:**

import java.util.\*;

class Main {

public static void main(String[] args) {

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

Scanner sc = new Scanner(System.in);

System.out.println("Enter number of string you want to insert in ArrayList");

int n = sc.nextInt();

System.out.println("Enter the String into ArrayList");

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

String s = sc.next();

array.add(s);

}

for (int i = 0; i < array.size(); i++) {

String word = array.get(i);

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

array.remove(i);

i--;

}

}

System.out.println("\n ArrayList after removal of even length Strings");

for (int i = 0; i < array.size(); i++) {

System.out.println(array.get(i));

}

sc.close();

}

}

**Output:**

Text, letter

Description automatically generated

**Q-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.**

**Sol:**

import java.util.\*;

class LabQ22 {

public static void main(String[] args) {

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

Scanner sc = new Scanner(System.in);

System.out.print("Enter number of string you want to insert in ArrayList: ");

int n = sc.nextInt();

System.out.println("Enter the String into ArrayList:- ");

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

String s = sc.next();

array.add(s);

}

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

String s = array.get(i);

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

array.set(i + 1, s);

}

System.out.println("\nArrayList after Switch: ");

for (int i = 0; i < array.size(); i++) {

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

}

sc.close();

}

}

**Output:**

Text

Description automatically generated

**Q-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**
* **…**

**If the lists do not contain the same number of elements, the remaining elements from the longer list should be placed consecutively at the end. For example, for a first list of (1, 2, 3, 4, 5) and a second list of (6, 7, 8, 9, 10, 11, 12), a call of alternate(list1, list2) should return a list containing (1, 6, 2, 7, 3, 8, 4, 9, 5, 10, 11, 12). Do not modify the parameter lists passed in.**

**Sol:**

import java.util.\*;

class LabQ23 {

public static ArrayList<Integer> alternate(ArrayList<Integer> a1, ArrayList<Integer> a2) {

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

int i = 0, j = 0;

while (i < a1.size() && j < a2.size()) {

ans.add(a1.get(i));

ans.add(a2.get(j));

i++;

j++;

}

while (i < a1.size()) {

ans.add(a1.get(i));

i++;

}

while (j < a2.size()) {

ans.add(a2.get(j));

j++;

}

return ans;

}

public static void main(String[] args) {

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

ArrayList<Integer> array2 = new ArrayList<Integer>();

Scanner sc = new Scanner(System.in);

System.out.println("Enter number of integer you want to insert in ArrayList: ");

int n1 = sc.nextInt();

System.out.println("Enter the integer into ArrayList1: ");

for (int i = 0; i < n1; i++) {

int x = sc.nextInt();

array.add(x);

}

System.out.println("Enter number of integer you want to insert in ArrayList: ");

int n2 = sc.nextInt();

System.out.println("Enter the integer into ArrayList1: ");

for (int i = 0; i < n2; i++) {

int x = sc.nextInt();

array2.add(x);

}

ArrayList<Integer> ans = alternate(array, array2);

System.out.println("\nArrayList after removal of even length Strings");

for (int i = 0; i < ans.size(); i++) {

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

}

sc.close();

}

}

**Output:**

Text, letter

Description automatically generated

**Q-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 textfield one and Textfield two .**

**When the button named “Exit” is clicked it will close the application.**

**Sol:**

import java.awt.event.\*;

import java.awt.\*;

public class LabQ24 extends Frame {

LabQ24() {

Label str = new Label("Enter String");

str.setBounds(20, 50, 80, 20);

Label result = new Label("Result");

result.setBounds(20, 80, 80, 20);

TextField stringInput = new TextField();

stringInput.setBounds(120, 50, 100, 20);

TextField resultOut = new TextField();

resultOut.setBounds(120, 80, 100, 20);

Button countVowel = new Button("Count Vowel");

countVowel.setBounds(20, 140, 100, 20);

countVowel.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent ae) {

String temp = stringInput.getText();

Integer count = 0;

temp.toLowerCase();

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

if (temp.charAt(i) == 'a' || temp.charAt(i) == 'e' || temp.charAt(i) == 'i' || temp.charAt(i) == 'o'

||

temp.charAt(i) == 'u') {

count++;

}

}

resultOut.setText(count.toString());

}

});

Button reset = new Button("Reset");

reset.setBounds(120, 140, 100, 20);

reset.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent ae) {

stringInput.setText("");

resultOut.setText("");

}

});

Button bhar = new Button("Exit");

bhar.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent ae) {

System.exit(0);

}

});

bhar.setBounds(220, 140, 100, 20);

add(str);

add(result);

add(stringInput);

add(resultOut);

add(countVowel);

add(reset);

add(bhar);

setTitle("LabQ24");

setSize(340, 250);

setLayout(null);

setVisible(true);

setResizable(true);

}

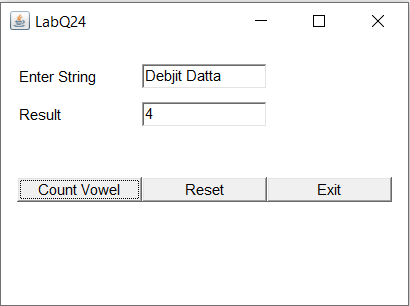
public static void main(String args[]) {

new LabQ24();

}

}

**Output:**



**Q-25. Create a database of employee with the following fields: -**

**🡪Name**

**🡪Code**

**🡪Designation**

**🡪Salary**

**(a)Write a java program to create GUI java application to take employee data from the TextFields and store in database using JDBC connectivity.**

**(b) Write a JDBC Program to retrieves all the records from employee database.**

**Sol:**

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.Statement;

class GuiDb implements ActionListener {

Frame f = new Frame();

Label l1 = new Label("NAME");

Label l2 = new Label("CODE");

Label l3 = new Label("DESIGNATION");

Label l4 = new Label("SALARY");

TextField t1 = new TextField();

TextField t2 = new TextField();

TextField t3 = new TextField();

TextField t4 = new TextField();

Button b1 = new Button("SAVE");

Button b2 = new Button("RESET");

Button b3 = new Button("EXIT");

GuiDb() {

l1.setBounds(50, 100, 100, 20);

l2.setBounds(50, 120, 100, 20);

l3.setBounds(50, 140, 100, 20);

l4.setBounds(50, 160, 100, 20);

t1.setBounds(150, 100, 150, 20);

t2.setBounds(150, 120, 150, 20);

t3.setBounds(150, 140, 150, 20);

t4.setBounds(150, 160, 150, 20);

b1.setBounds(75, 200, 70, 20);

b2.setBounds(155, 200, 50, 20);

b3.setBounds(215, 200, 50, 20);

f.add(l1);

f.add(l2);

f.add(l3);

f.add(l4);

f.add(t1);

f.add(t2);

f.add(t3);

f.add(t4);

f.add(b1);

f.add(b2);

f.add(b3);

f.setLayout(null);

f.setVisible(true);

f.setSize(400, 350);

}

public void actionPerformed(ActionEvent e) {

if (e.getSource() == b1) {

String name = t1.getText();

String code = t2.getText();

String desig = t3.getText();

String sal = t4.getText();

UpdateDisplay ud = new UpdateDisplay();

ud.updateDB(name, code, desig, sal);

}

if (e.getSource() == b2) {

t1.setText("");

t2.setText("");

t3.setText("");

t4.setText("");

}

if (e.getSource() == b3) {

UpdateDisplay ud = new UpdateDisplay();

ud.displayDB();

System.exit(0);

}

}

}

class UpdateDisplay {

void updateDB(String name, String code, String desig, String sal) {

String query = "insert into employee values(?,?,?,?)";

try {

Class.forName("com.mysql.jdbc.Driver");

Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/employees", "root", "");

PreparedStatement st = con.prepareStatement(query);

st.setString(1, name);

st.setString(2, code);

st.setString(3, desig);

st.setString(4, sal);

st.executeUpdate();

st.close();

con.close();

} catch (Exception e) {

System.out.println(e);

}

}

void displayDB() {

try {

Class.forName("com.mysql.jdbc.Driver");

Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/employees", "root", "");

Statement stmt = con.createStatement();

ResultSet rs = stmt.executeQuery("select \* from employee");

System.out.println("DATABASE: ");

while (rs.next()) {

System.out.println(

rs.getString(1) + " " + rs.getString(2) + " " + rs.getString(3) + " " + rs.getString(4));

}

con.close();

} catch (Exception e) {

System.out.println(e);

}

}

}

class LabQ25 {

public static void main(String[] args) {

GuiDb g = new GuiDb();

}

}

**Output:**

Graphical user interface, application, Word

Description automatically generated

**DATABASE**

Table

Description automatically generated with medium confidence