

**Practical No. 1:**

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

Name:

University Roll No:

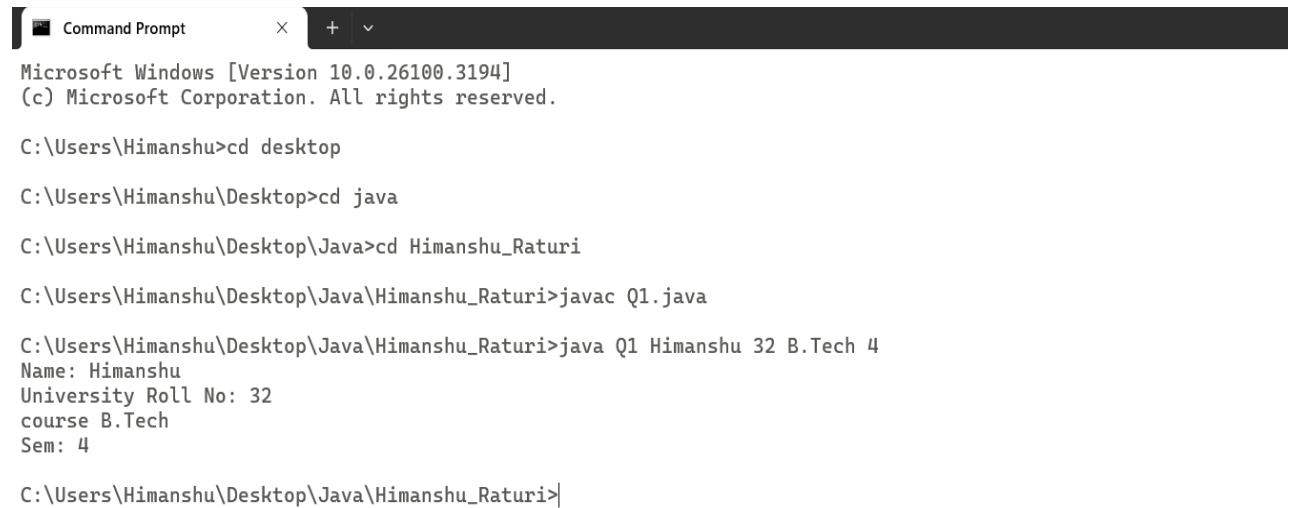
Course:

Semester:

**Source Code:**

```
public class Q1 {  
    public static void main(String[] args) {  
        String name = args[0];  
        int RollNo = Integer.parseInt(args[1]);  
        String course = args[2];  
        int sem = Integer.parseInt(args[3]);  
        System.out.println("Name: " + name);  
        System.out.println("University Roll No: " + RollNo);  
        System.out.println("course " + course);  
        System.out.println("Sem: " + sem);  
    }  
}
```

## Output:



```
Microsoft Windows [Version 10.0.26100.3194]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Himanshu>cd desktop

C:\Users\Himanshu\Desktop>cd java

C:\Users\Himanshu\Desktop\Java>cd Himanshu_Raturi

C:\Users\Himanshu\Desktop\Java\Himanshu_Raturi>javac Q1.java

C:\Users\Himanshu\Desktop\Java\Himanshu_Raturi>java Q1 Himanshu 32 B.Tech 4
Name: Himanshu
University Roll No: 32
course B.Tech
Sem: 4

C:\Users\Himanshu\Desktop\Java\Himanshu_Raturi>
```

## 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.util.Scanner;

//import java.lang.*;

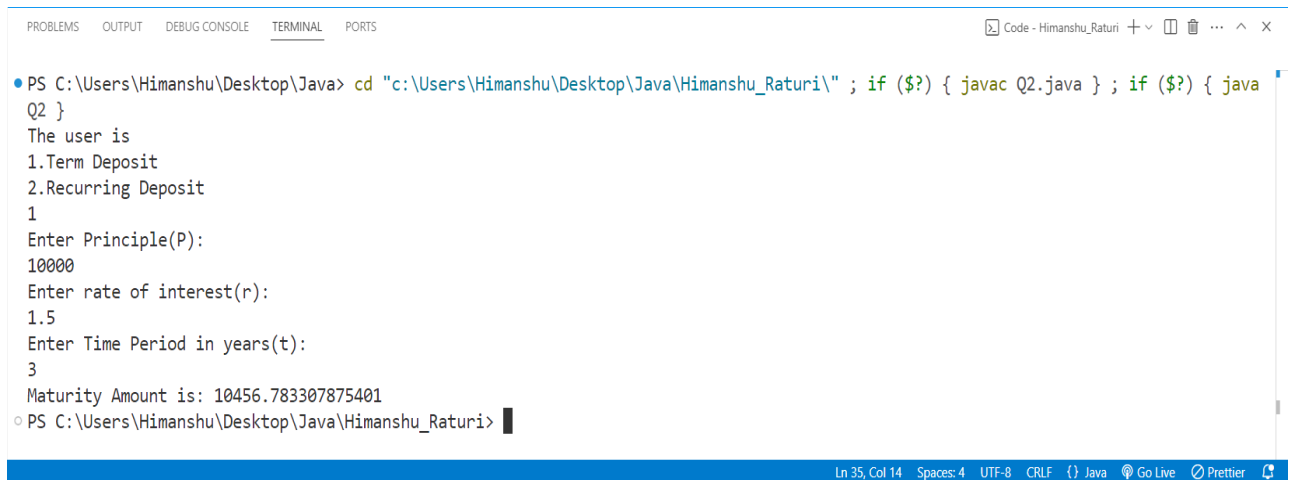
public class Q2
{
    public static void main(String args[])
    {
        System.out.println("The user is\n1.Term Deposit\n2.Recurring Deposit");
        Scanner sc = new Scanner(System.in);
        int Choice = sc.nextInt();
        switch(Choice)
        {
            case 1:
            {
                System.out.println("Enter Principle(P): ");
                int Princ = sc.nextInt();
                System.out.println("Enter rate of interest(r): ");
                float Rate = sc.nextFloat();
                System.out.println("Enter Time Period in years(t): ");
                int t = sc.nextInt();
                double Maturity = Princ*Math.pow(1+Rate/100,t);
```

```

        System.out.println("Maturity Amount is: " + Maturity);
        break;
    }
    case 2:
    {
        System.out.println("Enter Monthly installment(P): ");
        double p = sc.nextInt();
        System.out.println("Enter Rate of Interest(r): ");
        double r = sc.nextInt();
        System.out.println("Enter Time period(yrs): ");
        double t = sc.nextInt();
        double Maturity = p * t + p * t*(t + 1) / 2 * r / 100 * 1 / 12;
        System.out.println("Maturity is: " + Maturity);
        break;
    }
}
sc.close();
}
}

```

## Output:



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Code - Himanshu_Raturi + - [ ] ... ^ X
• PS C:\Users\Himanshu\Desktop\Java> cd "c:\Users\Himanshu\Desktop\Java\Himanshu_Raturi\" ; if ($?) { javac Q2.java } ; if ($?) { java
Q2 }
The user is
1.Term Deposit
2.Recurring Deposit
1
Enter Principle(P):
10000
Enter rate of interest(r):
1.5
Enter Time Period in years(t):
3
Maturity Amount is: 10456.783307875401
○ PS C:\Users\Himanshu\Desktop\Java\Himanshu_Raturi>
```

Ln 35, Col 14 Spaces: 4 UTF-8 CRLF {} Java Go Live Prettier

**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 Number 1: ");

        int num1 = sc.nextInt();

        System.out.println("Enter Number 2: ");

        int num2 = sc.nextInt();

        int sum1 = 0, sum2 = 0;

        for (int i = 1; i <= num1 / 2; i++)

        {

            if (num1 % i == 0)

            {

                sum1 += i;

            }

        }

        for (int i = 1; i <= num2 / 2; i++) {

            if (num2 % i == 0) {

                sum2 += i;

            }

        }

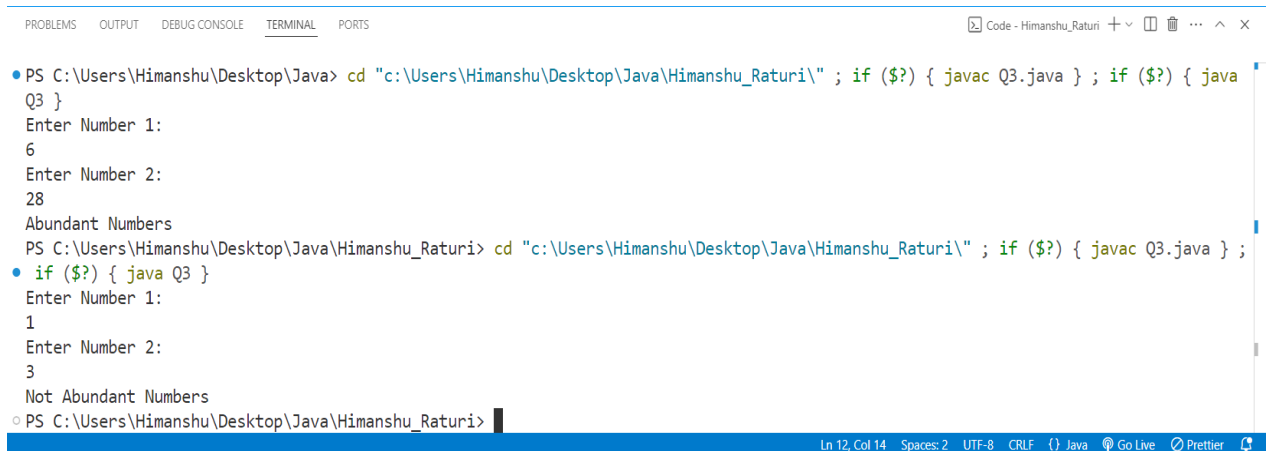
        if ((sum1 == num1) && (sum2 == num2)) {

            System.out.println("Abundant Numbers");

        } else {
```

```
        System.out.println("Not Abundant Numbers");  
    }  
    sc.close();  
}  
}
```

## Output:



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Code - Himanshu_Raturi + - [ ] ... ^ X

• PS C:\Users\Himanshu\Desktop\Java> cd "c:\Users\Himanshu\Desktop\Java\Himanshu_Raturi\" ; if ($?) { javac Q3.java } ; if ($?) { java
Q3 }
Enter Number 1:
6
Enter Number 2:
28
Abundant Numbers
PS C:\Users\Himanshu\Desktop\Java\Himanshu_Raturi> cd "c:\Users\Himanshu\Desktop\Java\Himanshu_Raturi\" ; if ($?) { javac Q3.java } ;
• if ($?) { java Q3 }
Enter Number 1:
1
Enter Number 2:
3
Not Abundant Numbers
○ PS C:\Users\Himanshu\Desktop\Java\Himanshu_Raturi>

Ln 12, Col 14 Spaces: 2 UTF-8 CRLF {} Java Go Live Prettier
```



**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:**

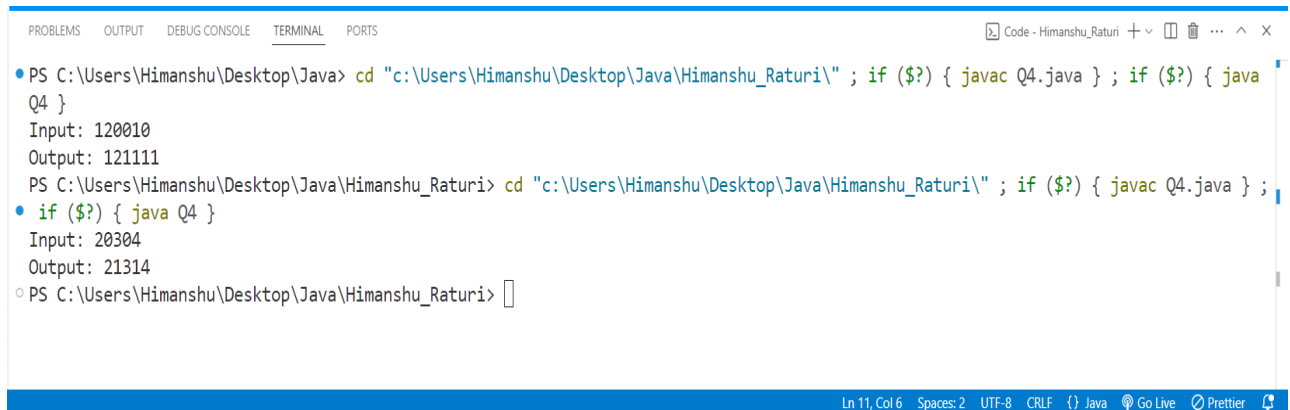
```
import java.util.Scanner;

public class Q4 {

    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Input: ");
        int num = sc.nextInt();
        if (num == 0) {
            num = 1;
        }
        int x = 0;
        while (num > 0) {
            int val = num % 10;
            if (val == 0) {
                val = 1;
            }
            x = x * 10 + val;
            num = num / 10;
        }
        int ans = 0;
        while (x > 0) {
            int val = x % 10;
            ans = ans * 10 + val;
            x = x / 10;
        }
    }
}
```

```
}  
System.out.print("Output: " + ans);  
sc.close();  
}  
}
```

## Output:



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Code - Himanshu_Raturi + - [ ] ... ^ x

• PS C:\Users\Himanshu\Desktop\Java> cd "c:\Users\Himanshu\Desktop\Java\Himanshu_Raturi\" ; if ($?) { javac Q4.java } ; if ($?) { java
Q4 }
Input: 120010
Output: 121111
PS C:\Users\Himanshu\Desktop\Java\Himanshu_Raturi> cd "c:\Users\Himanshu\Desktop\Java\Himanshu_Raturi\" ; if ($?) { javac Q4.java } ;
• if ($?) { java Q4 }
Input: 20304
Output: 21314
○ PS C:\Users\Himanshu\Desktop\Java\Himanshu_Raturi> 
```

Ln 11, Col 6 Spaces: 2 UTF-8 CRLF { } Java Go Live Prettier

### 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 java5 {

    public static void main(String[] args) {

        {

            Scanner in=new Scanner(System.in);

            int n;

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

            n=in.nextInt();

            int arr[]=new int[n];

            int temp;

            System.out.println("Enter the element in the array ");

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

            {

                arr[i]=in.nextInt();

            }

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

            {

                if(i%2==0)

                {

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

                    {

                        temp=arr[i];

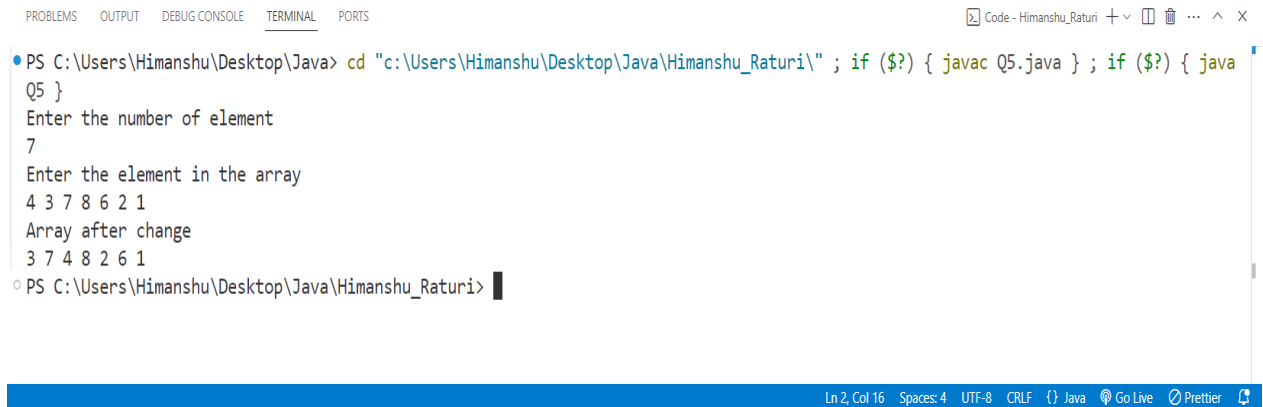
                        arr[i]=arr[i+1];
```

```

        arr[i+1]=temp;
    }
}
else
{
    if(arr[i]<arr[i+1])
    {
        temp=arr[i];
        arr[i]=arr[i+1];
        arr[i+1]=temp;
    }
}
}
System.out.println("Array after change ");
for(int i=0;i<n;i++)
{
    System.out.print(arr[i]+" ");
}
in.close();
}
}
}

```

## Output:



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Code - Himanshu_Raturi + - [ ] ... ^ X

• PS C:\Users\Himanshu\Desktop\Java> cd "c:\Users\Himanshu\Desktop\Java\Himanshu_Raturi" ; if ($?) { javac Q5.java } ; if ($?) { java
Q5 }
Enter the number of element
7
Enter the element in the array
4 3 7 8 6 2 1
Array after change
3 7 4 8 2 6 1
○ PS C:\Users\Himanshu\Desktop\Java\Himanshu_Raturi>

Ln 2, Col 16 Spaces: 4 UTF-8 CRLF {} Java Go Live Prettier
```

**Practical No. 6:** The problem to rearrange positive and negative numbers in an array .

Method: This approach moves all negative numbers to the beginning and positive numbers to the end but changes the order of appearance of the elements of the array.

**Source Code:**

```
package CODES.Java.Himanshu_Raturi;

import java.util.Scanner;

public class Q6
{
    public static void main(String args[])
    {
        Scanner sc = new Scanner(System.in);

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

        int n = sc.nextInt();

        int arr[] = new int[n];

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

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

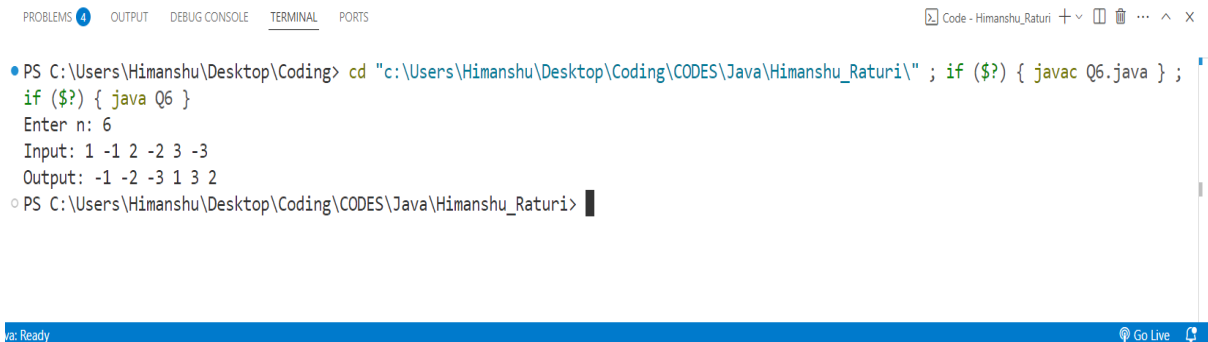
        int first = 0;

        for(int i = 0; i < n ; i++)
        {
            if(arr[i] < 0 )
            {
                int temp = arr[i];
                arr[i] = arr[first];
                arr[first] = temp;
                first++;
            }
        }
    }
}
```

```
System.out.print("Output: ");  
for(int i = 0 ; i < n ; i++)  
{  
    System.out.print(arr[i] + " ");  
}  
sc.close();  
}  
}
```



## Output:



```
PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL PORTS Code - Himanshu_Raturi + - [ ] ... ^ X

• PS C:\Users\Himanshu\Desktop\Coding> cd "c:\Users\Himanshu\Desktop\Coding\CODES\Java\Himanshu_Raturi\" ; if ($?) { javac Q6.java } ;
if ($?) { java Q6 }
Enter n: 6
Input: 1 -1 2 -2 3 -3
Output: -1 -2 -3 1 3 2
○ PS C:\Users\Himanshu\Desktop\Coding\CODES\Java\Himanshu_Raturi>

vs: Ready Go Live
```

**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:**

```
package CODES.Java.Himanshu_Raturi;

import java.util.Scanner;

public class Q7 {

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

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

        int rows=sc.nextInt();

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

        int columns=sc.nextInt();

        int arr[][]=new int[rows][columns];

        System.out.println("Enter the elements in the array ");

        for(int i=0;i<rows;i++)
        {
            for(int j=0;j<columns;j++)
            {
                arr[i][j]=sc.nextInt();

            }
        }

        for(int i=0;i<rows;i++)
        {
            int min=arr[i][0];

            int colindex=0;

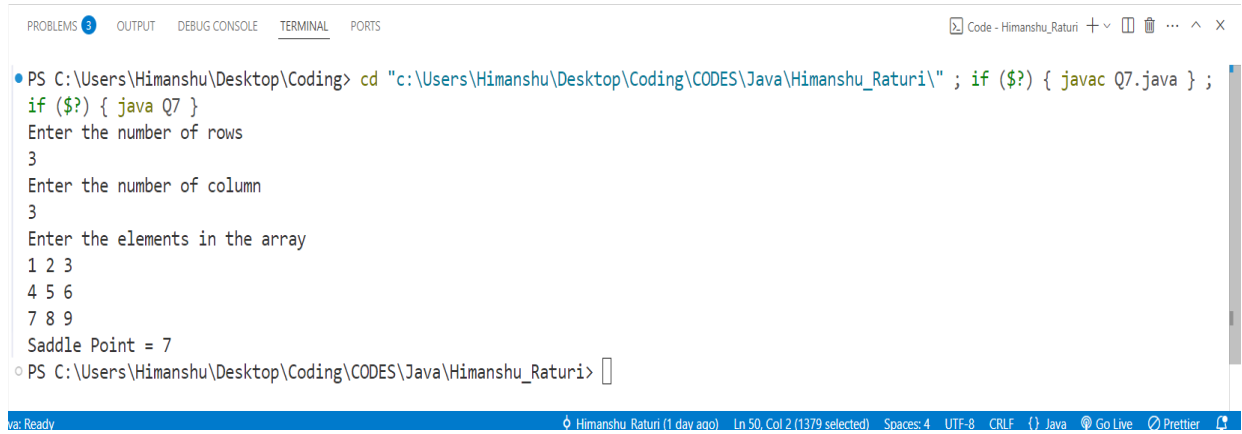
            for(int j=0;j<columns;j++)
            {
```

```

        if(arr[i][j]<min)
        {
            min=arr[i][j];
            colindex=j;
        }
    }
    int max=arr[0][colindex];
    for(int k=0;k<rows;k++)
    {
        if(arr[k][colindex]>max)
        {
            max=arr[k][colindex];
        }
    }
    if(min==max)
    {
        System.out.println("Saddle Point = "+min );
        break;
    }
}
sc.close();
}
}

```

## OUTPUT:



```
PS C:\Users\Himanshu\Desktop\Coding> cd "c:\Users\Himanshu\Desktop\Coding\CODES\Java\Himanshu_Raturi\" ; if ($?) { javac Q7.java } ;  
if ($?) { java Q7 }  
Enter the number of rows  
3  
Enter the number of column  
3  
Enter the elements in the array  
1 2 3  
4 5 6  
7 8 9  
Saddle Point = 7  
PS C:\Users\Himanshu\Desktop\Coding\CODES\Java\Himanshu_Raturi>
```

**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:**

```
package CODES.Java.Himanshu_Raturi;

import java.util.Scanner;

public class Q8
{
    public static void main(String args[])
    {
        String str;

        Scanner sc = new Scanner(System.in);

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

        str = sc.nextLine();

        //String str = new String("01101111010");

        int count = 0 ;

        for(int i =0 ; i < str.length() - 1; i++)
        {
            if(str.charAt(i) == '0' && str.charAt(i+1) == '1')
            {
                count++;
            }
        }

        System.out.println(count);

        sc.close();
    }
}
```

## Output:



The screenshot shows a VS Code terminal window with the following content:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Code - Himanshu_Raturi + - [ ] ... ^ X

• PS C:\Users\Himanshu\Desktop\Coding> cd "c:\Users\Himanshu\Desktop\Coding\CODES\Java\Himanshu_Raturi\" ; if ($?) { javac Q8.java } ;
if ($?) { java Q8 }
Enter a string: 01101111010
3
• PS C:\Users\Himanshu\Desktop\Coding\CODES\Java\Himanshu_Raturi> 
```

The status bar at the bottom indicates: Ready, Himanshu\_Raturi (1 day ago), Ln 14, Col 10, Spaces: 4, UTF-8, CRLF, Java, Go Live, Prettier.

**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

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

**Source code:**

```
package CODES.Java.Himanshu_Raturi;
```

```
import java.util.Scanner;
```

```
public class Q9_Bank
```

```
{
```

```
String name , address ;
```

```
int accno;
```

```
double balance;
```

```
void setName(String name)
```

```
{
```

```
    this.name = name;
```

```
}
```

```
void setAddress(String address)
```

```
{
```

```
    this.address = address;
```

```
}
```

```
void setAccno(int i)
```

```
{
```

```

        this.accno = accno + 1001 + i;
    }

    void setBalance(double balance)
    {
        this.balance = balance;
    }

    int getacc()
    {
        return accno;
    }

    void display() {
        System.out.println("Name: " + name + "\n" +
            "Address: " + address + "\n" +
            "Account Number: " + accno + "\n" +
            "Balance:INR " + balance);
    }

    void deposit(int amt)
    {
        balance += amt;

        System.out.println("INR " + amt+" has been successfully deposited.\n" + "Total amount is:
"+balance);
    }

    void withdraw(int amt)
    {
        if(amt > balance)
        {
            System.out.println("Insufficient Balance availble.");
        }else

```



```

    {
        balance -= amt;

        System.out.println(amt+" has been successfully withdrawn.\n" + "Total amount is:
"+balance);
    }
}

void changeAddress(String add)
{
    System.out.println("Address has been successfully changed from "+ address + " to " +
add);
    address = add;

}

public static void main(String args[])
{
    int n;
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter Number of depositors: ");
    n = sc.nextInt();
    Q9_Bank depositors[] = new Q9_Bank[n];
    for(int i = 0 ; i < n ; i++)
    {
        depositors[i] = new Q9_Bank();
    }
    for(int i = 0 ; i < n; i++)
    {
        String name , address ;
        double balance;

        System.out.println("Enter Details of " + " user:- " +(1001+i)+":");
        System.out.print("Enter Name: ");

```

```

        name = sc.next();
        depositors[i].setName(name);
        System.out.print("Enter Address: ");
        address = sc.next();
        depositors[i].setAddress(address);
        depositors[i].setAccno(i);
        System.out.print("Enter Balance: ");
        balance = sc.nextDouble();
        depositors[i].setBalance(balance);
    }
    int choice;

    int accno;
    System.out.print("Enter account number to operate: ");
    accno = sc.nextInt();
    do
    {
        System.out.println("Press:\n" + "1 to Deposit Money\n" + "2 to withdraw money\n" + "3 to
Change addres\n"+ "4 to display Information\n"+ "5 to exit.");
        choice = sc.nextInt();
        switch(choice)
        {
            case 1:
                {
                    int amt;
                    System.out.println("Enter amount to deposit: ");
                    amt = sc.nextInt();
                    for(int i = 0 ; i < n ;i++)
                    {

```

```

        if(depositors[i].getacc() == accno)
        {
            depositors[i].deposit(amt);
            break;
        }
    }
    break;
}

case 2:
{
    int amt;
    System.out.println("Enter amount to Withdraw: ");
    amt = sc.nextInt();
    for(int i = 0 ; i < n ;i++)
    {
        if(depositors[i].getacc() == accno)
        {
            depositors[i].withdraw(amt);
            break;
        }
    }
    break;
}

case 3:
{
    String add;
    System.out.println("Enter New address: ");
    add = sc.next();
    for(int i = 0 ; i < n ;i++)

```

```

        {
            if(depositors[i].getacc() == accno)
            {
                depositors[i].changeAddress(add);
                break;
            }
        }
        break;
    }
case 4:
    {
        for(int i = 0 ; i < n ;i++)
        {
            if(depositors[i].getacc() == accno)
            {
                depositors[i].display();
                break;
            }
        }
        break;
    }
}

}while(choice != 5);
System.out.println("Exiting System.Thank you.....");
sc.close();
}
}

```

**Output:**

```

PS C:\Users\Himanshu\Desktop\Coding> cd "c:\Users\Himanshu\Desktop\Coding\CODES\Java\Himanshu_Raturi\" ; if ($?) { javac Q9_Bank.java } ; if ($?) { java Q9_Bank }
Enter Number of depositors: 2
Enter Details of user:- 1001:
Enter Name: Himanshu
Enter Address: Rishikesh
Enter Balance: 5000
Enter Details of user:- 1002:
Enter Name: Bhaumik
Enter Address: haridwar
Enter Balance: 10000
Enter account number to operate: 1002
Press:
1 to Deposit Money
2 to withdraw money
3 to Change address
4 to display Information
5 to exit.
1
Enter amount to deposit:
2000
INR 2000 has been successfully deposited.
Total amount is: 12000.0
Press:
1 to Deposit Money
2 to withdraw money
3 to Change address

```

```
2 to withdraw money
3 to Change addres
4 to display Information
5 to exit.
2
Enter amount to Withdraw:
5000
5000 has been successfully withdrawn.
Total amount is: 7000.0
Press:
1 to Deposit Money
2 to withdraw money
3 to Change addres
4 to display Information
5 to exit.
3
Enter New address:
dehradun
Address has been successfully changed from haridwar to dehradun
Press:
1 to Deposit Money
2 to withdraw money
3 to Change addres
4 to display Information
5 to exit.
4
Name: Bhaumik
```

```
PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL PORTS
Code - Himanshu_Raturi + - [ ] [ ] ... ^ x

3 to Change address
4 to display Information
5 to exit.
3
Enter New address:
dehradun
Address has been successfully changed from haridwar to dehradun
Press:
1 to Deposit Money
2 to withdraw money
3 to Change address
4 to display Information
5 to exit.
4
Name: Bhaumik
Address: dehradun
Account Number: 1002
Balance:INR 7000.0
Press:
1 to Deposit Money
2 to withdraw money
3 to Change address
4 to display Information
5 to exit.
5
Exiting System.Thank you.....
PS C:\Users\Himanshu\Desktop\Coding\CODING\Java\Himanshu_Raturi>
a: Ready Not Committed Yet Ln 1, Col 3 Tab Size: 4 UTF-8 CRLF {} Java Go Live Prettier
```

**Practical No. 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:**

```
package CODES.Java.Himanshu_Raturi;

import java.util.Scanner;

public class Q10_word_example {

    private String strdata;

    public Q10_word_example(String str)

    {

        int l = str.length();

        if(str.charAt(l - 1) == '.' || str.charAt(l - 1) == '?' || str.charAt(l - 1) == '!')

        {

            strdata = str;

        }else

        {

            System.out.println("Enter a valid string ending with ? , . , !.");

        }

    }

    boolean isVowel(String word)

    {

        if ((word.charAt(0) == 'A' || word.charAt(0) == 'E' || word.charAt(0) == 'I'

            || word.charAt(0) == 'O' || word.charAt(0) == 'U')

            &&

            (word.charAt(word.length() - 1) == 'A' || word.charAt(word.length() - 1) == 'E'
```

```

    || word.charAt(word.length() - 1) == 'T' || word.charAt(word.length() - 1) == 'O'
    || word.charAt(word.length() - 1) == 'U'))
{
    return true;
} else
{
    return false;
}
}

void countword()
{
    int count = 0;
    String[] token = strdata.trim().split("\\s+");
    for(String word : token)
    {
        if (isVowel(word))
        {
            count++;
        }
    }

    System.out.println("Words starting and ending with Vowels are: " + count);
}

void placeWord()
{
    String[] token = strdata.trim().split("\\s+");
    StringBuilder vowelWords = new StringBuilder();
    StringBuilder otherWords = new StringBuilder();
    for(String word : token)

```



```

{
    if (isVowel(word))
    {
        vowelWords.append(word + " ");
    }else
    {
        otherWords.append(word + " ");
    }
}

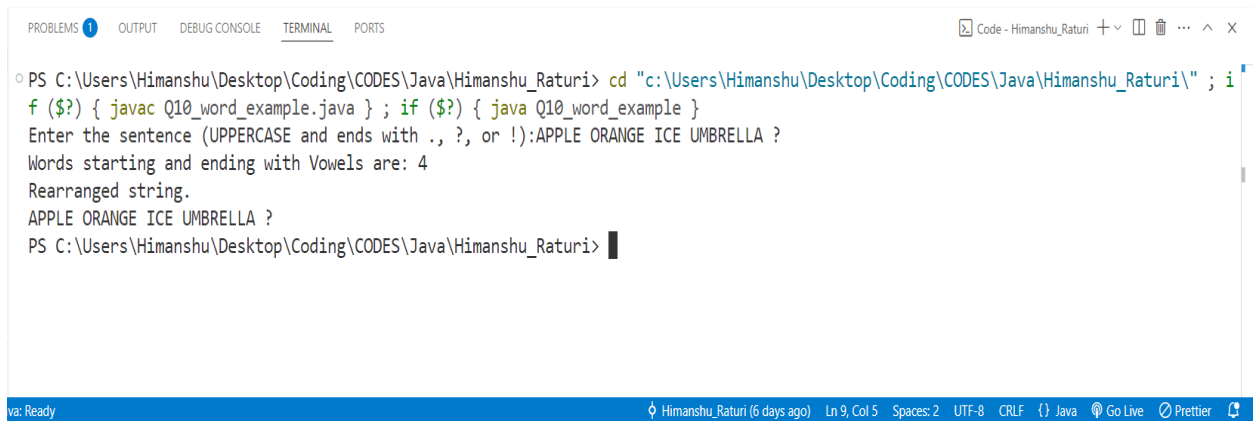
System.out.println("Rearranged string.");
System.out.print(vowelWords);
System.out.println(otherWords + " ");

}

public static void main(String args[])
{
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter the sentence (UPPERCASE and ends with ., ?, or !):");
    String str;
    str = sc.nextLine();
    Q10_word_example obj = new Q10_word_example(str);
    obj.countword();
    obj.placeWord();
    sc.close();
}
}

```

## OUTPUT:



```
PS C:\Users\Himanshu\Desktop\Coding\CODES\Java\Himanshu_Raturi> cd "c:\Users\Himanshu\Desktop\Coding\CODES\Java\Himanshu_Raturi\" ; if ($?) { javac Q10_word_example.java } ; if ($?) { java Q10_word_example }
Enter the sentence (UPPERCASE and ends with ., ?, or !):APPLE ORANGE ICE UMBRELLA ?
Words starting and ending with Vowels are: 4
Rearranged string.
APPLE ORANGE ICE UMBRELLA ?
PS C:\Users\Himanshu\Desktop\Coding\CODES\Java\Himanshu_Raturi>
```

**Practical No. 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 .

**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.

**Source Code:**

```
package CODES.Java.Himanshu_Raturi;

import java.util.Scanner;

public class Q11_ArrayDemo {

    void arrayFunc(int arr[] , int key)

    {

        int l = arr.length;

        System.out.println("Pair of element whose sum is " + key + " are:");

        for(int i = 0 ; i < l ; i++)

        {

            for(int j = i+1 ; j < l ; j++)

            {

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

                if(sum == key)

                {

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

                }

            }

        }

    }

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

    {

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

```

int i = 0 , j = 0 , k = 0;
while(i < p && j < q)
{
    if(A[i] < B[j])
    {
        temp[k++] = A[i++];
    }else
    {
        temp[k++] = B[j++];
    }
}
while(i < p)
{
    temp[k++] = A[i++];
}
while(j < q)
{
    temp[k++] = B[j++];
}
k=0;
for(int x = 0 ; x < p ; x++)
{
    A[x] = temp[k++];
}
for(int y = 0 ; y < q ; y++)
{
    B[y] = temp[k++];
}
System.out.println("Sorted Arrays: ");

```

```

        System.out.print("A: [");
        for(int x = 0 ; x < p ; x++)
        {
            System.out.print(A[x] + ", ");
            if(x == p-1)
            {
                System.out.println("]" + "");
            }
        }
        System.out.print("B: [");
        for(int y = 0 ; y < q ; y++)
        {
            System.out.print(B[y] + ", ");
            if(y == q-1)
            {
                System.out.println("]" + "");
            }
        }

    }

    public static void main(String args[])
    {
        Scanner sc = new Scanner(System.in);
        int p , q ;
        Q11_ArrayDemo obj = new Q11_ArrayDemo();
        System.out.println("First arrayFunc Function.");
        System.out.print("Enter size of Array A[: ");
        p = sc.nextInt();
        int A[] = new int[p];
    }

```

```

System.out.print("Enter element in Array A[: ");
for(int i = 0 ; i < p ; i++)
{
    A[i] = sc.nextInt();
}
System.out.print("Enter Target: ");
int key = sc.nextInt();
obj.arrayFunc(A, key);
System.out.println("Second arrayFunc Function.");
System.out.print("Enter size of array A[: ");
int pa = sc.nextInt();
System.out.print("Enter size of array B[: ");
q = sc.nextInt();
int Aa[] = new int[pa];
int B[] = new int[q];
System.out.print("Enter element in Array A[: ");
for(int i = 0 ; i < pa ; i++)
{
    Aa[i] = sc.nextInt();
}
System.out.print("Enter element in Array B[: ");
for(int i = 0 ; i < q ; i++)
{
    B[i] = sc.nextInt();
}
obj.arrayFunc(Aa, pa , B , q);
sc.close();
}
}

```

## OUTPUT:



```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS
Code - Himanshu_Raturi + - [ ] ... ^ x

• PS C:\Users\Himanshu\Desktop\Coding\CODES\Java\Himanshu_Raturi> cd "c:\Users\Himanshu\Desktop\Coding\CODES\Java\Himanshu_Raturi\" ; i
f ($?) { javac Q11_ArrayDemo.java } ; if ($?) { java Q11_ArrayDemo }
First arrayFunc Function.
Enter size of Array A[]: 7
Enter element in Array A[]: 4 6 5 -10 8 5 20
Enter Target: 10
Pair of element whose sum is 10 are:
4 + 6 = 10
5 + 5 = 10
-10 + 20 = 10
Second arrayFunc Function.
Enter size of array A[]: 6
Enter size of array B[]: 3
Enter element in Array A[]: 1 5 6 7 8 10
Enter element in Array B[]: 2 4 9
Sorted Arrays:
A: [1, 2, 4, 5, 6, 7, ]
B: [8, 9, 10, ]
PS C:\Users\Himanshu\Desktop\Coding\CODES\Java\Himanshu_Raturi>
```

**Practical No. 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.

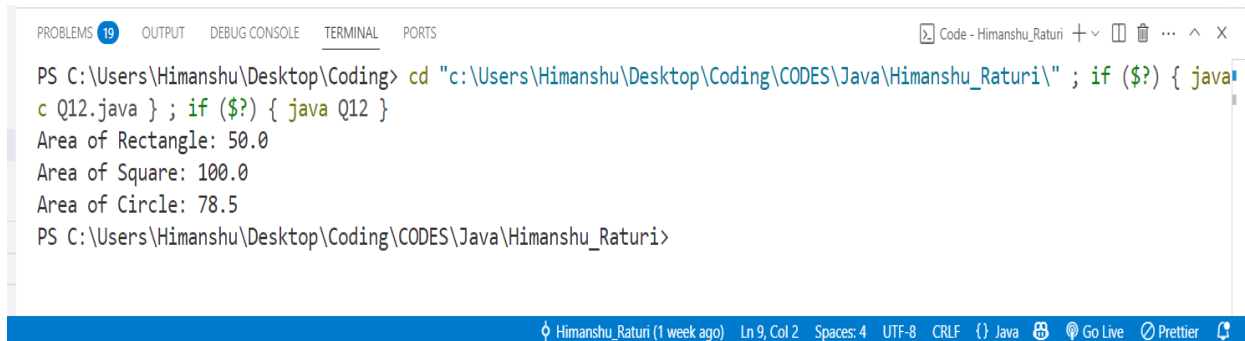
**Source Code:**

```
abstract class Shape
{
    abstract void rectangleArea(double l, double b);
    abstract void squareArea(double side);
    abstract void circleArea(double radius);
}
class Area extends Shape
{
    void rectangleArea(double l, double b)
    {
        double area = l * b;
        System.out.println("Area of Rectangle: " + area);
    }
    void squareArea(double side)
    {
        double area = side * side;
        System.out.println("Area of Square: " + area);
    }
    void circleArea(double radius)
    {
        double area = 3.14 * radius * radius;
        System.out.println("Area of Circle: " + area);
    }
}
```



```
}  
public class Q12 {  
    public static void main(String args[])  
    {  
        Shape obj = new Area();  
        obj.rectangleArea(10, 5);  
        obj.squareArea(10);  
        obj.circleArea(5);  
    }  
}
```

## OUTPUT:



```
PS C:\Users\Himanshu\Desktop\Coding> cd "c:\Users\Himanshu\Desktop\Coding\CODES\Java\Himanshu_Raturi\" ; if ($?) { java -c Q12.java } ; if ($?) { java Q12 }
Area of Rectangle: 50.0
Area of Square: 100.0
Area of Circle: 78.5
PS C:\Users\Himanshu\Desktop\Coding\CODES\Java\Himanshu_Raturi>
```

**Practical No. 13:** 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), abstract 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:**

```
package CODES.Java.Himanshu_Raturi;

import java.util.Scanner;

abstract class Temperature
{
    double temp;

    void setTempData(double t)
    {
        temp = t;
    }

    abstract void changeTemp();
}

class Fahrenheit extends Temperature
{
    double ctemp;

    void changeTemp()
    {
        ctemp = 5.0/9 * (temp - 32);

        System.out.println("Temperature in celcius : " + ctemp + " °C");
    }
}

class Celsius extends Temperature
```

```

{
    double ftemp;
    void changeTemp()
    {
        ftemp = (9.0/5) * temp + 32;
        System.out.println("Temperature in Fahrenheit: " + ftemp + " °C");
    }
}

public class Q13 {
    public static void main(String args[])
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter temperature in Fahrenheit: ");
        double ftemp = sc.nextDouble();
        Temperature f = new Fahrenheit();
        f.setTempData(ftemp);
        f.changeTemp();

        System.out.print("Enter temperature in Celcius: ");
        double ctemp = sc.nextDouble();
        Temperature c = new Celsius();
        c.setTempData(ctemp);
        c.changeTemp();
        sc.close();
    }
}

```

## OUTPUT:



```
PROBLEMS 19 OUTPUT DEBUG CONSOLE TERMINAL PORTS Code - Himanshu_Raturi + - [ ] [ ] ... ^ X

PS C:\Users\Himanshu\Desktop\Coding> cd "c:\Users\Himanshu\Desktop\Coding\CODES\Java\Himanshu_Raturi\" ; if ($?) { java
c Q13.java } ; if ($?) { java Q13 }
• Enter temperature in Fahrenheit: 25
  Temperature in Celsius: -3.89 °C
  Enter temperature in Celcius: -3.89
  Temperature in Fahrenheit: 25.00 F
○ PS C:\Users\Himanshu\Desktop\Coding\CODES\Java\Himanshu_Raturi> █

Not Committed Yet Ln 27, Col 71 Spaces: 4 UTF-8 CRLF {} Java [ ] [ ] Go Live [ ] Prettier [ ]
```

**Practical No. 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.

Volume of cone =  $(1/3)\pi r^2 h$

Volume of hemisphere =  $(2/3)\pi r^3$

Volume of cylinder =  $\pi r^2 h$

**Source Code:**

```
import java.util.Scanner;

interface Shape
{
    void volume();
}

class Cone implements Shape
{
    double radius , height;

    Cone(double radius , double height)
    {
        this.radius = radius;
        this.height = height;
    }

    public void volume()
    {
        double vol = (1.0/3) * Math.PI * Math.pow(radius, 2.0) * height;
        System.out.printf("Volume of Cone is: %2f meter cube\n",vol);
    }
}

class Hemisphere implements Shape
{
    double radius;
```

```

Hemisphere(double radius)
{
    this.radius = radius;
}

public void volume()
{
    double vol = (2.0/3)* Math.PI * Math.pow(radius, 3.0);
    System.out.printf("Volume of Hemisphere is: %2f meter cube\n",vol);
}
}

class Cylinder implements Shape
{
    double radius , height;
    Cylinder(double radius , double height)
    {
        this.radius = radius;
        this.height = height;
    }
    public void volume()
    {
        double vol = Math.PI * Math.pow(radius, 2.0) * height;
        System.out.printf("Volume of Cylinder is: %2f meter cube\n",vol);
    }
}

public class Q14 {
    public static void main(String args[])
    {
        Scanner sc = new Scanner(System.in);
    }
}

```

```
System.out.print("Enter Radius of Cone: ");  
double cradius = sc.nextDouble();  
System.out.print("Enter Height of cone: ");  
double cheight = sc.nextDouble();  
Shape cone = new Cone(cradius, cheight);  
cone.volume();
```

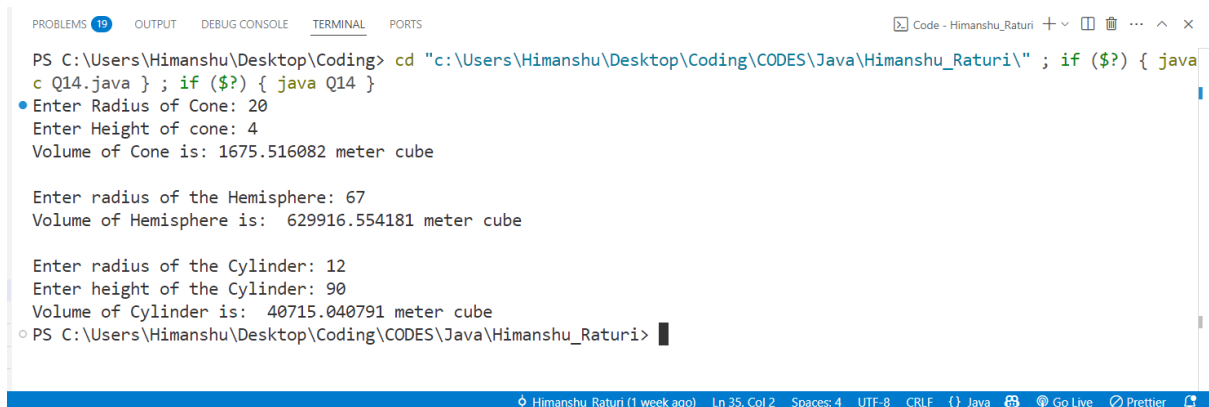
```
System.out.print("\nEnter radius of the Hemisphere: ");  
double hemisphereRadius = sc.nextDouble();  
Shape hemisphere = new Hemisphere(hemisphereRadius);  
hemisphere.volume();
```

```
System.out.print("\nEnter radius of the Cylinder: ");  
double cylinderRadius = sc.nextDouble();  
System.out.print("Enter height of the Cylinder: ");  
double cylinderHeight = sc.nextDouble();  
Shape cylinder = new Cylinder(cylinderRadius, cylinderHeight);  
cylinder.volume();  
sc.close();
```

```
}  
}
```



## OUTPUT:



```
PROBLEMS 19 OUTPUT DEBUG CONSOLE TERMINAL PORTS
Code - Himanshu_Raturi + - [ ] ... ^ x

PS C:\Users\Himanshu\Desktop\Coding> cd "c:\Users\Himanshu\Desktop\Coding\CODES\Java\Himanshu_Raturi\" ; if ($?) { java
c Q14.java } ; if ($?) { java Q14 }
• Enter Radius of Cone: 20
Enter Height of cone: 4
Volume of Cone is: 1675.516082 meter cube

Enter radius of the Hemisphere: 67
Volume of Hemisphere is: 629916.554181 meter cube

Enter radius of the Cylinder: 12
Enter height of the Cylinder: 90
Volume of Cylinder is: 40715.040791 meter cube
○ PS C:\Users\Himanshu\Desktop\Coding\CODES\Java\Himanshu_Raturi>

Himanshu_Raturi (1 week ago) Ln 35, Col 2 Spaces: 4 UTF-8 CRLF { } Java [ ] Go Live [ ] Prettier [ ]
```

**Practical No. 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:**

**student package**

```
package student;

public class Student
{
    int Studentid;
    String name;
    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 package**

```
package course;

public class Course
{
    int courseId;
    String courseName;
    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 package**

```
package Enrollement;

import student.Student;
import course.Course;

public class Enrollment {
    Student std;
    Course crs;
    public Enrollment(Student student, Course course) {
        std = student;
    }
}
```

```

        crs = course;
    }

    public String getEnrollmentInfo() {
        return "Enrollment Details:\n" + std.getStudentInfo() + "\n" + crs.getCourseInfo();
    }
}

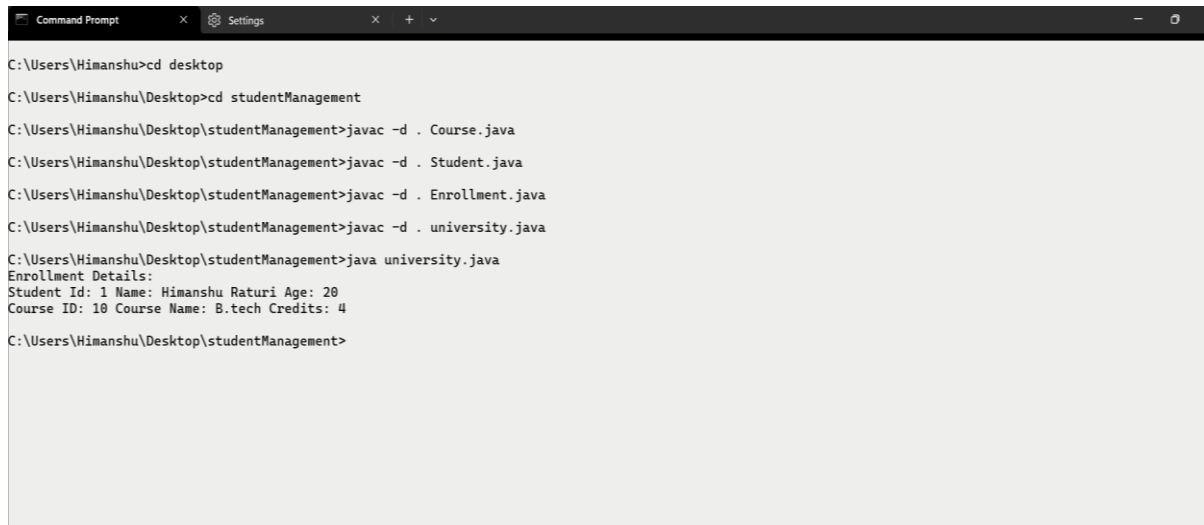
default package

import student.Student;
import Enrollement.Enrollment;
import course.Course;

public class university
{
    public static void main(String[] args)
    {
        Student s1 = new Student(1 , "Himanshu Raturi" , 20);
        Course c1 = new Course(10 , "B.tech" , 4);
        Enrollment e1 = new Enrollment(s1,c1);
        System.out.println(e1.getEnrollmentInfo());
    }
}

```

## OUTPUT:



```
Command Prompt
C:\Users\Himanshu>cd desktop
C:\Users\Himanshu\Desktop>cd studentManagement
C:\Users\Himanshu\Desktop\studentManagement>javac -d . Course.java
C:\Users\Himanshu\Desktop\studentManagement>javac -d . Student.java
C:\Users\Himanshu\Desktop\studentManagement>javac -d . Enrollment.java
C:\Users\Himanshu\Desktop\studentManagement>javac -d . university.java
C:\Users\Himanshu\Desktop\studentManagement>java university.java
Enrollment Details:
Student Id: 1 Name: Himanshu Raturi Age: 20
Course ID: 10 Course Name: B.tech Credits: 4
C:\Users\Himanshu\Desktop\studentManagement>
```

**Practical No. 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.Scanner;

class myNameException extends Exception
{
    public myNameException(String s)
    {
        super(s);
    }
}

class myIdException extends Exception
{
    public myIdException(String s)
    {
        super(s);
    }
}

class myDeptIdException extends Exception
{
    public myDeptIdException(String s)
    {
        super(s);
    }
}
```

```

public class Q16 {

    public static void main(String args[])
    {

        Scanner sc = new Scanner(System.in);

        try
        {
            int id , dept_Id;

            String name;

            System.out.print("Enter Employee ID (2001 - 5001): ");

            id = sc.nextInt();

            if(id < 2001 || id > 5001)
            {
                throw new myIdException("Error! Enter Valid Id between 2001 to 5001.");
            }

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

            name = sc.next();

            if(!Character.isUpperCase(name.charAt(0)))
            {
                throw new myNameException("Error! Enter Valid name.");
            }

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

            dept_Id = sc.nextInt();

            if(dept_Id < 1 || dept_Id > 5)
            {
                throw new myDeptIdException("Error! Enter valid Department Id (1 - 5).");
            }

            System.out.println("Detail of employee: ");

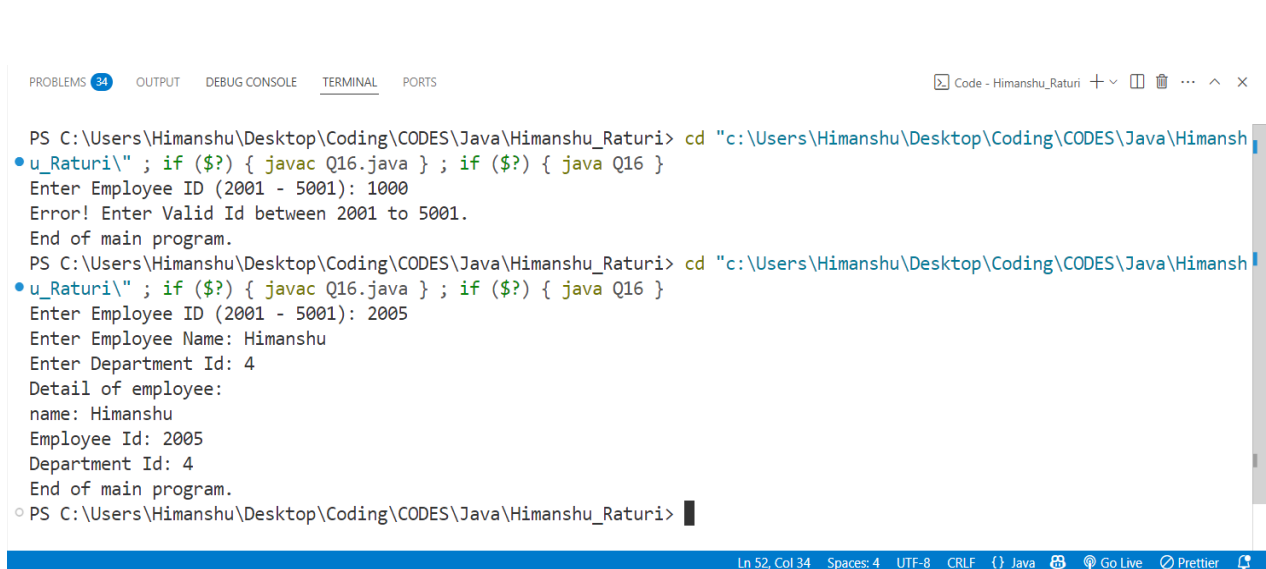
            System.out.println("name: " + name);
        }
    }
}

```

```
        System.out.println("Employee Id: " + id);
        System.out.println("Department Id: " + dept_Id);
    } catch (Exception e)
    {
        System.out.println(e.getMessage());
    }
    System.out.println("End of main program.");
    sc.close();
}
}
```



## OUTPUT:



```
PS C:\Users\Himanshu\Desktop\Coding\CODES\Java\Himanshu_Raturi> cd "c:\Users\Himanshu\Desktop\Coding\CODES\Java\Himanshu_Raturi\" ; if ($?) { javac Q16.java } ; if ($?) { java Q16 }
Enter Employee ID (2001 - 5001): 1000
Error! Enter Valid Id between 2001 to 5001.
End of main program.
PS C:\Users\Himanshu\Desktop\Coding\CODES\Java\Himanshu_Raturi> cd "c:\Users\Himanshu\Desktop\Coding\CODES\Java\Himanshu_Raturi\" ; if ($?) { javac Q16.java } ; if ($?) { java Q16 }
Enter Employee ID (2001 - 5001): 2005
Enter Employee Name: Himanshu
Enter Department Id: 4
Detail of employee:
name: Himanshu
Employee Id: 2005
Department Id: 4
End of main program.
PS C:\Users\Himanshu\Desktop\Coding\CODES\Java\Himanshu_Raturi>
```

**Practical No. 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  $n^p$ . 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.Scanner;

class MyException extends Exception
{
    public MyException(String message)
    {
        super(message);
    }
}

class Mycalculator
{
    int n , p;
    void power(int n , int p)
    {
        try
        {
            if(n < 0 || p < 0)
            {
                throw new MyException("n and p should be non-negative.");
            }
            else if (n == 0 || p == 0)
            {
                throw new MyException("Both n and p are zero.");
            }
            System.out.println(Math.pow(n,p));
        }
    }
}
```

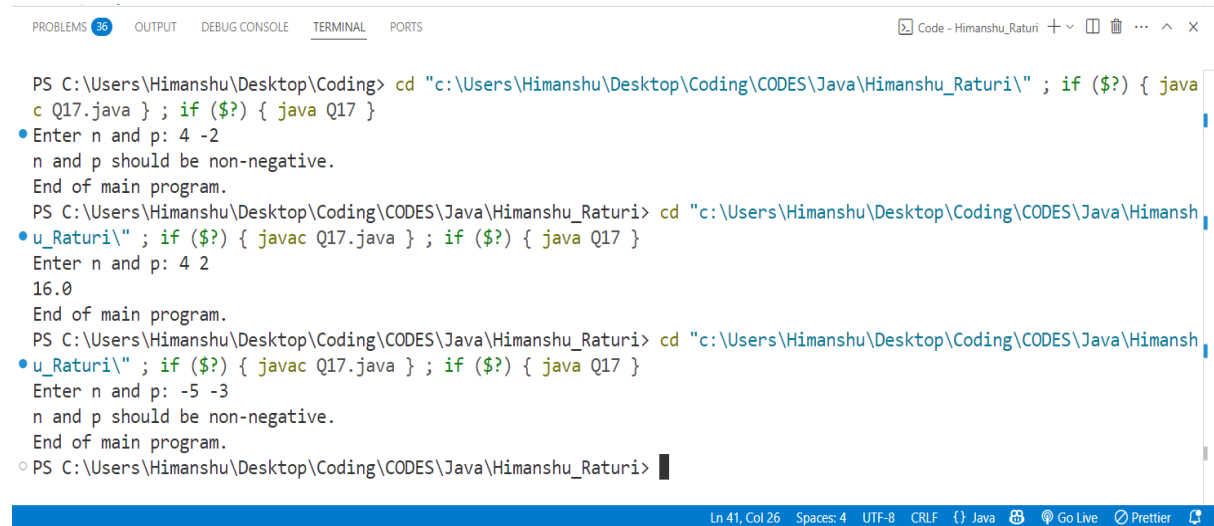
```

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

public class Q17 {
    public static void main(String args[])
    {
        Mycalculator obj = new Mycalculator();
        int n,p;
        System.out.print("Enter n and p: ");
        Scanner sc = new Scanner(System.in);
        n = sc.nextInt();
        p = sc.nextInt();
        obj.power(n, p);
        System.out.println("End of main program.");
    }
}

```

## OUTPUT:



```
PROBLEMS 36 OUTPUT DEBUG CONSOLE TERMINAL PORTS Code - Himanshu_Raturi + - [ ] ... ^ X

PS C:\Users\Himanshu\Desktop\Coding> cd "c:\Users\Himanshu\Desktop\Coding\CODES\Java\Himanshu_Raturi\" ; if ($?) { java
c Q17.java } ; if ($?) { java Q17 }
• Enter n and p: 4 -2
n and p should be non-negative.
End of main program.
PS C:\Users\Himanshu\Desktop\Coding\CODES\Java\Himanshu_Raturi> cd "c:\Users\Himanshu\Desktop\Coding\CODES\Java\Himansh
u_Raturi\" ; if ($?) { javac Q17.java } ; if ($?) { java Q17 }
• Enter n and p: 4 2
16.0
End of main program.
PS C:\Users\Himanshu\Desktop\Coding\CODES\Java\Himanshu_Raturi> cd "c:\Users\Himanshu\Desktop\Coding\CODES\Java\Himansh
u_Raturi\" ; if ($?) { javac Q17.java } ; if ($?) { java Q17 }
• Enter n and p: -5 -3
n and p should be non-negative.
End of main program.
○ PS C:\Users\Himanshu\Desktop\Coding\CODES\Java\Himanshu_Raturi> █

Ln 41, Col 26 Spaces: 4 UTF-8 CRLF {} Java [ ] Go Live [ ] Prettier [ ]
```

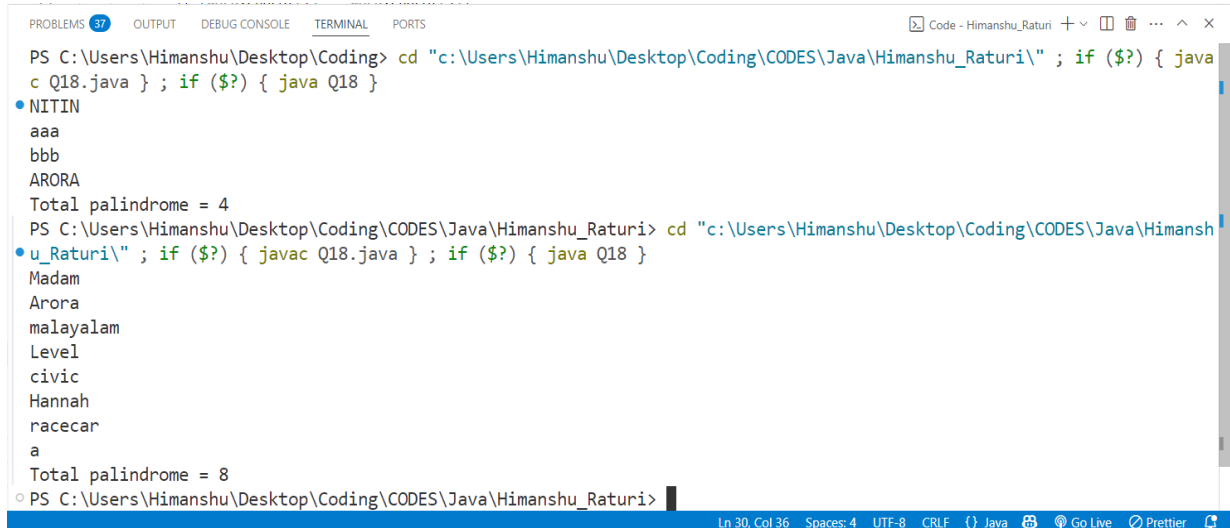
**Practical No. 18:** Write a java file handling program to count and display the number of palindromes present in a text file "myfile.txt".

**Source Code:**

```
import java.util.*;
import java.io.*;
public class Q18
{
    static boolean isPalindrome(String word)
    {
        word = word.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 IOException
    {
        int count = 0;
        File file = new File("C:\\Users\\Himanshu\\Desktop\\Myfile.txt");
        Scanner sc = new Scanner(file);
        while (sc.hasNext())
        {
```

```
String word=sc.next().replaceAll("[^a-zA-Z]", "");
if (isPalindrome(word))
{
    System.out.println(word);
    count++;
}
}
sc.close();
System.out.println("Total palindrome = " + count);
}
}
```

## OUTPUT:



```
PS C:\Users\Himanshu\Desktop\Coding> cd "c:\Users\Himanshu\Desktop\Coding\CODES\Java\Himanshu_Raturi\" ; if ($?) { javac Q18.java } ; if ($?) { java Q18 }
• NITIN
aaa
bbb
ARORA
Total palindrome = 4
PS C:\Users\Himanshu\Desktop\Coding\CODES\Java\Himanshu_Raturi> cd "c:\Users\Himanshu\Desktop\Coding\CODES\Java\Himanshu_Raturi\" ; if ($?) { javac Q18.java } ; if ($?) { java Q18 }
• u_Raturi\
Madam
Arora
malayalam
Level
civic
Hannah
racecar
a
Total palindrome = 8
PS C:\Users\Himanshu\Desktop\Coding\CODES\Java\Himanshu_Raturi>
```

**Practical No. 19:** Write a program MultiThreads that creates two threads-one 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 CSthread extends Thread{

    public void run()
    {
        int x = 1;
        while(true)
        {
            System.out.println("X: " + x);
            try
            {
                Thread.sleep(500);
            } catch (InterruptedException e)
            {
                System.out.println(e);
            }
            x++;
        }
    }
}

class ITthread extends Thread{

    public void run()
    {
        int y = 1;
        while(true)
        {
            System.out.println("Y: " + y);
```



```

        try
        {
            Thread.sleep(500);
        } catch (InterruptedException e)
        {
            System.out.println(e);
        }
        y++;
    }
}

}

public class Q19 {
    public static void main(String args[])
    {
        CSthread t1 = new CSthread();
        ITthread t2 = new ITthread();
        t1.setName("CSThread");
        t2.setName("ITThread");
        System.out.println(t1.getName());
        System.out.println(t2.getName());
        t1.start();
        t2.start();
    }
}

```

## OUTPUT:

```
PROBLEMS 41 OUTPUT DEBUG CONSOLE TERMINAL PORTS Code - Himanshu_Raturi + - [ ] ... ^ X
PS C:\Users\Himanshu\Desktop\Coding> cd "c:\Users\Himanshu\Desktop\Coding\CODES\Java\Himanshu_Raturi\" ; if ($?) { java
c Q19.java } ; if ($?) { java Q19 }
* CThread
ITThread
X: 1
Y: 1
X: 2
Y: 2
Y: 3
X: 3
Y: 4
X: 4
Y: 5
X: 5
Y: 6
X: 6
Y: 7
X: 7
Y: 8
X: 8
Y: 9
X: 9
Y: 10
X: 10
PS C:\Users\Himanshu\Desktop\Coding\CODES\Java\Himanshu_Raturi>
```

**Practical No. 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 = 2;
    public void produce() throws Exception
    {
        int v = 0;
        while(true)
        {
            synchronized(this)
            {
                while(list.size() == capacity)
                {
                    wait();
                }
                System.out.println("Producer is going to produce..." +v);
                list.add(v++);
                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 obj = new Pc1();
    A(Pc1 obj)
    {
        this.obj = obj;
    }
    public void run()
    {
        try
        {
            obj.produce();
        } catch (Exception e)
        {
            System.out.println(e);
        }
    }
}

```

```

    }
}
class B extends Thread
{
    Pc1 obj = new Pc1();
    B(Pc1 obj)
    {
        this.obj = obj;
    }
    public void run()
    {
        try
        {
            obj.consume();
        } catch (Exception e)
        {
            System.out.println(e);
        }
    }
}
public class Q20 {
    public static void main(String args[]) throws InterruptedException
    {
        Pc1 obj = new Pc1();
        A t1 = new A(obj);
        B t2 = new B(obj);
        t1.start();
        t2.start();
    }
}

```

## OUTPUT:

```
PS C:\Users\Himanshu\Desktop\Coding\CODES\Java\Himanshu_Raturi> java Q20
Producer is going to produce...0
Consumer is going to consume: 0
Producer is going to produce...1
Consumer is going to consume: 1
Producer is going to produce...2
Producer is going to produce...3
Consumer is going to consume: 2
Consumer is going to consume: 3
Producer is going to produce...4
Producer is going to produce...5
Consumer is going to consume: 4
Producer is going to produce...6
Consumer is going to consume: 5
Producer is going to produce...7
Consumer is going to consume: 6
Producer is going to produce...8
Consumer is going to consume: 7
Producer is going to produce...9
Consumer is going to consume: 8
Producer is going to produce...10
Consumer is going to consume: 9
Producer is going to produce...11
Consumer is going to consume: 10
Producer is going to produce...12
Consumer is going to consume: 11
Consumer is going to consume: 12
Producer is going to produce...13
PS C:\Users\Himanshu\Desktop\Coding\CODES\Java\Himanshu_Raturi>
```

**Practical No. 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 Q21 {

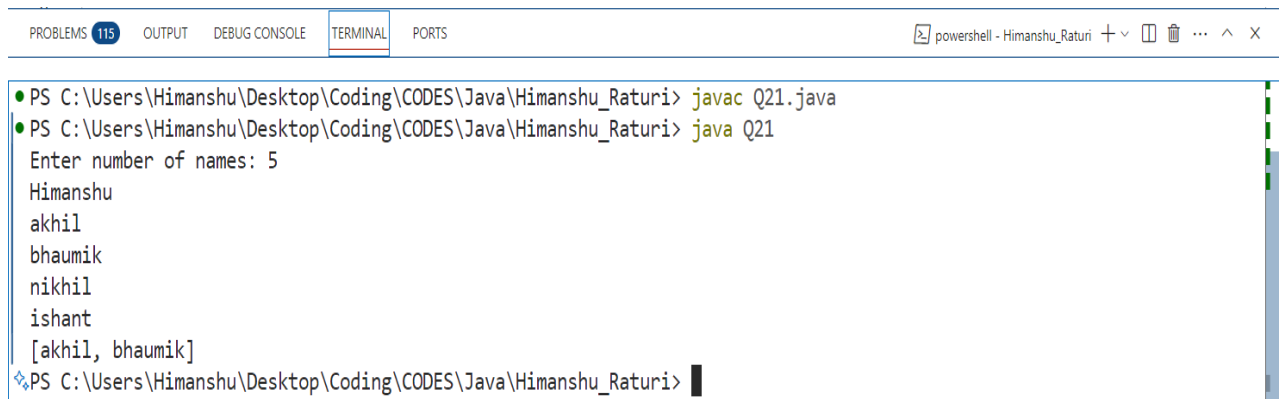
    static void removeEvenLength(ArrayList<String> l1)

    {
        Iterator<String> itr = l1.iterator();
        while (itr.hasNext())
        {
            String str = itr.next();
            if (str.length() % 2 == 0) {
                itr.remove();
            }
        }
    }

    public static void main(String args[]) {

        ArrayList<String> l1 = new ArrayList<String>();
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter number of names: ");
        int n = sc.nextInt();
        for(int i = 0 ; i < n ; i++)
        {
            String str = sc.next();
            l1.add(str);
        }
        removeEvenLength(l1);
        System.out.println(l1);
        sc.close();
    }
}
```

## OUTPUT:



The screenshot shows a terminal window with the following content:

```
PS C:\Users\Himanshu\Desktop\Coding\CODES\Java\Himanshu_Raturi> javac Q21.java
PS C:\Users\Himanshu\Desktop\Coding\CODES\Java\Himanshu_Raturi> java Q21
Enter number of names: 5
Himanshu
akhil
bhaumik
nikhil
ishant
[akhil, bhaumik]
PS C:\Users\Himanshu\Desktop\Coding\CODES\Java\Himanshu_Raturi>
```

The terminal window has a title bar that says "powershell - Himanshu\_Raturi". The tabs at the top are "PROBLEMS 115", "OUTPUT", "DEBUG CONSOLE", "TERMINAL", and "PORTS".



**Practical No. 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.*;

public class Q22 {

    static void swapPair(ArrayList<String> list)

    {
        for(int i = 0 ; i < list.size() - 1; i+=2)
        {
            String temp = list.get(i);
            list.set(i,list.get(i+1));
            list.set(i+1 , temp);
        }
    }

    public static void main(String[] args)
    {
        ArrayList<String> list = new ArrayList<String>();
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter number of values: ");
        int n = sc.nextInt();
        for(int i = 0 ; i < n ; i++)
        {
            String str = sc.next();
            list.add(str);
        }
        swapPair(list);
        System.out.println(list);
    }
}
```

## OUTPUT:

```
PS C:\Users\Himanshu\Desktop\Coding\CODES\Java\Himanshu_Raturi> javac Q22.java
PS C:\Users\Himanshu\Desktop\Coding\CODES\Java\Himanshu_Raturi> java Q22
Enter number of values: 6
four score and seven years ago
[score, four, seven, and, ago, years]
PS C:\Users\Himanshu\Desktop\Coding\CODES\Java\Himanshu_Raturi>
```

**Practical No. 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.

**Source Code:**

```
import java.util.*;

public class Q23 {

    static LinkedList<Integer> alternate(LinkedList<Integer> list1 , LinkedList<Integer> list2)
    {
        LinkedList<Integer> list3 = new LinkedList<Integer>();
        Iterator<Integer> itr1 = list1.iterator();
        Iterator<Integer> itr2 = list2.iterator();
        while(itr1.hasNext() || itr2.hasNext())
        {
            if(itr1.hasNext())
            {
                list3.add(itr1.next());
            }
            if(itr2.hasNext())
            {
                list3.add(itr2.next());
            }
        }
        return list3;
    }

    public static void main(String args[])
    {
        LinkedList<Integer> list1 = new LinkedList<Integer>();
        LinkedList<Integer> list2 = new LinkedList<Integer>();
        LinkedList<Integer> list3 = new LinkedList<Integer>();
        Scanner sc = new Scanner(System.in);
```

```
System.out.print("Enter number of element for List1: ");
int n = sc.nextInt();
System.out.print("Enter element for List 1:");
for(int i = 0 ; i< n ; i++)
{
    list1.add(sc.nextInt());
}
System.out.print("Enter number of element for List2: ");
int m = sc.nextInt();
System.out.print("Enter element for List 2:");
for(int i = 0 ; i< m ; i++)
{
    list2.add(sc.nextInt());
}
list3 = alternate(list1,list2);
System.out.println(list3);
}
}
```

## OUTPUT:

```
PS C:\Users\Himanshu\Desktop\Coding\CODES\Java\Himanshu_Raturi> javac Q23.java
PS C:\Users\Himanshu\Desktop\Coding\CODES\Java\Himanshu_Raturi> java Q23
Enter number of element for List1: 5
Enter element for List 1:1 2 3 4 5
Enter number of element for List2: 7
Enter element for List 2:6 7 8 9 10 11 12
[1, 6, 2, 7, 3, 8, 4, 9, 5, 10, 11, 12]
PS C:\Users\Himanshu\Desktop\Coding\CODES\Java\Himanshu_Raturi>
```

arning Ln 46, Col 1 (1497 selected) Spaces: 4 UTF-8 CRLF {} Java Go Live Prettier

**Practical No. 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 closed the application.

**Source Code:**

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

class Q24 extends JFrame implements ActionListener
{
    JTextField tf1 , tf2;

    Q24()
    {
        tf1 = new JTextField(20);
        tf2 = new JTextField(20);
        JLabel l1 = new JLabel("Enter String");
        JLabel l2 = new JLabel("Result");
        JButton b1 = new JButton("CountVowel");
        JButton b2 = new JButton("Reset");
        JButton b3 = new JButton("Exit");
        setLayout(new FlowLayout(FlowLayout.CENTER,30,10));
        add(l1); add(tf1);
        add(l2); add(tf2);
        add(b1); add(b2);
        add(b3);
        b1.addActionListener(this);
        b2.addActionListener(this);
        b3.addActionListener(this);
    }
}
```

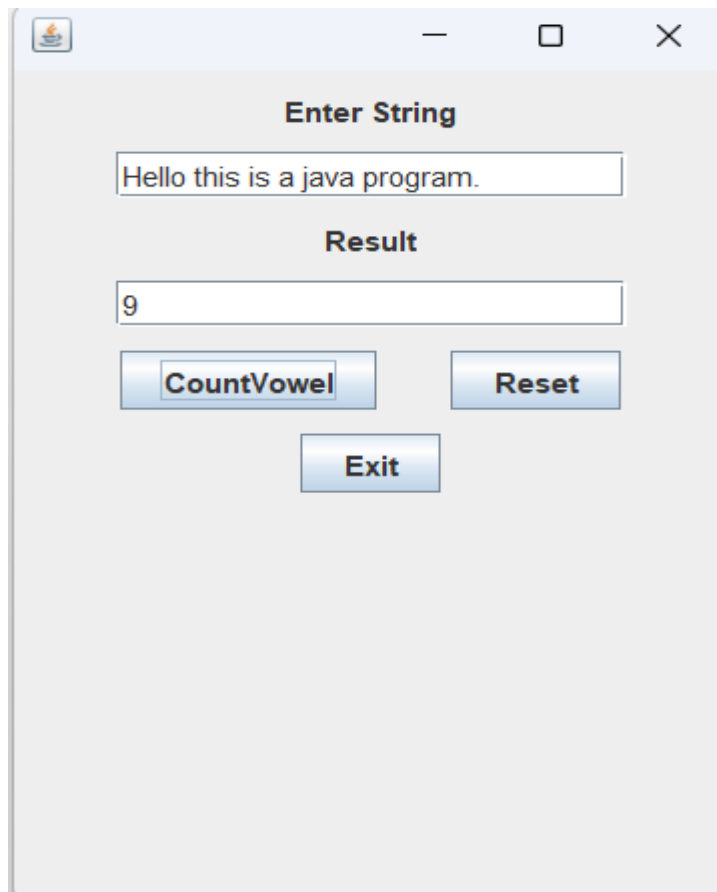
```

public void actionPerformed(ActionEvent e)
{
    String s = tf1.getText();
    int l = s.length();
    int vCount = 0;
    for(int i = 0 ; i < l ; i++)
    {
        char x = s.charAt(i);
        if(x == 'a' || x == 'e' || x == 'i' || x == 'o' || x == 'u')
            vCount++;
    }
    String a = e.getActionCommand();
    if(a.equals("CountVowel"))
    {
        tf2.setText(Integer.toString(vCount));
    }else if(a.equals("Reset"))
    {
        tf1.setText("");
        tf2.setText("");
    }else
    {
        System.exit(0);
    }
}

public static void main(String[] args) {
    Q24 d = new Q24();
    d.setSize(300,400);
    d.setVisible(true);
}
}

```

**OUTPUT:**



The screenshot shows a Java Swing window with a title bar containing a logo, a minus sign, a maximize button, and a close button. The window has a light gray background. At the top, the text "Enter String" is centered. Below it is a text input field containing the string "Hello this is a java program.". Underneath the input field, the text "Result" is centered. Below "Result" is another text input field containing the number "9". At the bottom of the window, there are three buttons: "CountVowel" and "Reset" are side-by-side, and "Exit" is centered below them. All buttons have a blue gradient and a 3D effect.



**Practical No. 25:** Design a JavaFX-based movie ticket booking system where users can select a movie, choose a showtime, and specify the number of tickets they wish to purchase.

- (a) The system should calculate the total cost based on the movie and the number of tickets, and display it in the UI.
- (b) Users can confirm their booking with a button, which will show a confirmation message
- (c) Additionally, a reset button should allow users to clear all selections and start over.
- (d) Provide an exit button to close the application.

**Source Code:**

```
import javafx.application.Application;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.ComboBox;
import javafx.scene.control.Label;
import javafx.scene.control.RadioButton;
import javafx.scene.control.TextArea;
import javafx.scene.control.ToggleGroup;
import javafx.scene.layout.HBox;
import javafx.scene.layout.VBox;
import javafx.stage.Stage;

public class Q25 extends Application {
    private ComboBox<String> movies;
    private ComboBox<String> tickets;
    private ToggleGroup group;
    private TextArea ta;
    private Stage primaryStage;

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

```

public void start(Stage stage) throws Exception {
    primaryStage = stage;
    ta = new TextArea();
    Label l1 = new Label("Select Movie");
    Button b1 = new Button("Book Ticket");
    Button resetBtn = new Button("Reset");
    Button exitBtn = new Button("Exit");
    movies = new ComboBox<>();
    movies.getItems().addAll("The Shawshank Redemption", "The Godfather", "Astitva",
    "Mother India");
    tickets = new ComboBox<>();
    tickets.getItems().addAll("1", "2", "3", "4");
    group = new ToggleGroup();
    RadioButton r1 = new RadioButton("10:00am");
    RadioButton r2 = new RadioButton("1:00pm");
    RadioButton r3 = new RadioButton("4:00pm");
    RadioButton r4 = new RadioButton("7:00pm");
    r1.setToggleGroup(group);
    r2.setToggleGroup(group);
    r3.setToggleGroup(group);
    r4.setToggleGroup(group);
    HBox movieSelection = new HBox(10, l1, movies);
    HBox timeSelection = new HBox(10, r1, r2, r3, r4);
    HBox ticketSelection = new HBox(10, tickets);
    HBox buttons = new HBox(10, b1, resetBtn, exitBtn);
    VBox root = new VBox(15, movieSelection, timeSelection, ticketSelection, buttons, ta);
    Scene sc = new Scene(root, 600, 300);
    stage.setScene(sc);
}

```

```

stage.setTitle("Movie Ticket Booking");
stage.show();
b1.setOnAction(this::bookTicket);
resetBtn.setOnAction(this::resetForm);
exitBtn.setOnAction(this::exitApp);
}

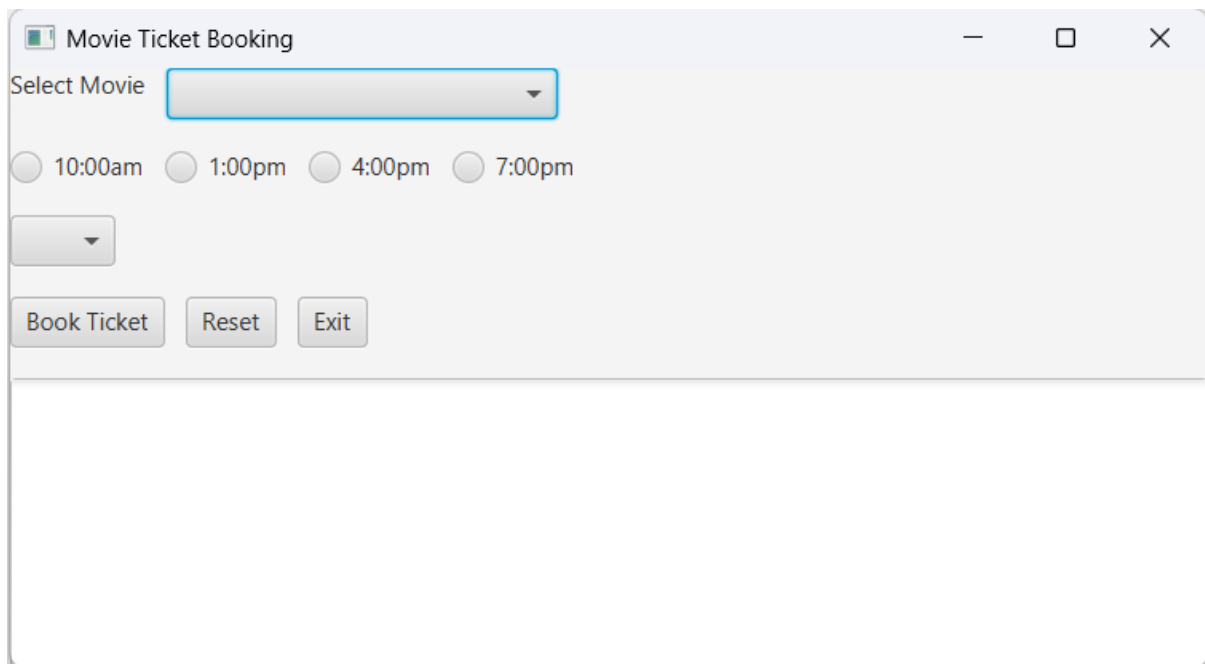
private void bookTicket(ActionEvent e) {
    String s = null;
    if (((RadioButton) group.getSelectedToggle()) != null) {
        s = ((RadioButton) group.getSelectedToggle()).getText();
    } try {
        int tt = Integer.parseInt(tickets.getValue());
        int charges = 100 * tt;
        ta.setText("Movie Name: " + movies.getValue()
            + "\nShow Time: " + s
            + "\nYou selected " + tickets.getValue() + " ticket(s)"
            + "\nTotal Cost: ₹" + charges);
    } catch (Exception ex) {
        ta.setText("Please make sure all fields are selected correctly.");
    }
}

private void resetForm(ActionEvent e) {
    movies.getSelectionModel().clearSelection();
    tickets.getSelectionModel().clearSelection();
    group.selectToggle(null);
    ta.clear();
}

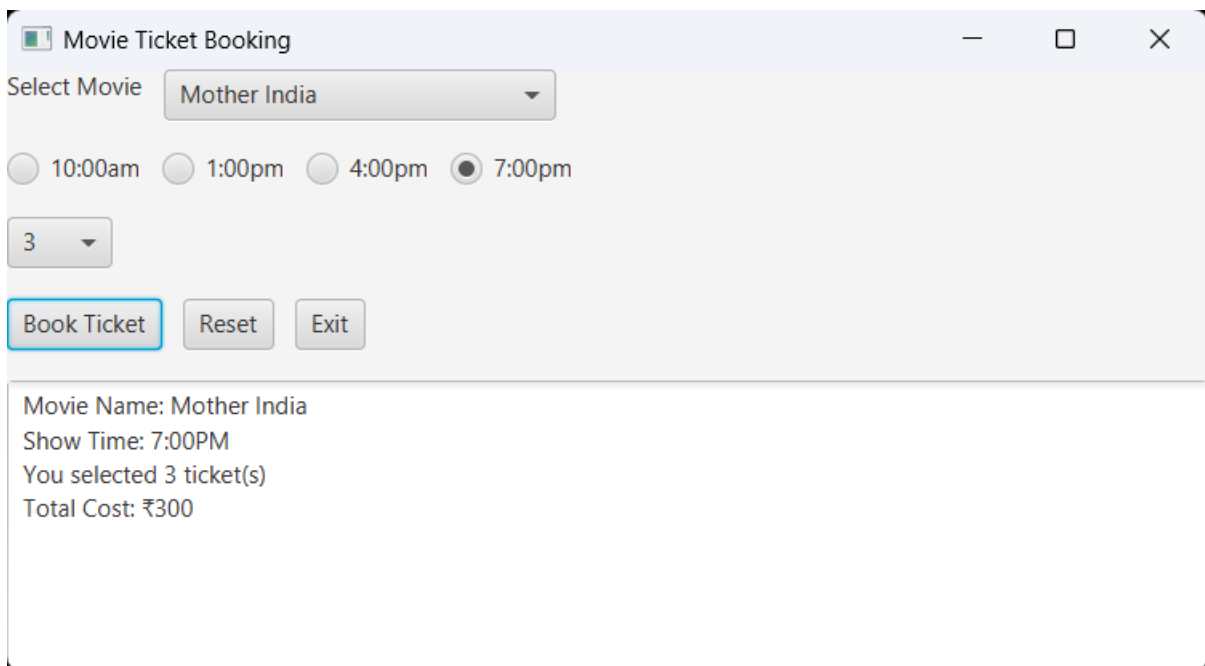
private void exitApp(ActionEvent e) {
    primaryStage.close();
}
}

```

## Output:



The screenshot shows a window titled "Movie Ticket Booking" with standard Windows window controls (minimize, maximize, close). The interface includes a "Select Movie" dropdown menu, four radio buttons for showtimes (10:00am, 1:00pm, 4:00pm, 7:00pm), a dropdown for the number of tickets, and three buttons: "Book Ticket", "Reset", and "Exit". The "Book Ticket" button is highlighted with a blue border. The bottom section of the window is empty.



The screenshot shows the same "Movie Ticket Booking" window after a booking. The "Select Movie" dropdown now displays "Mother India". The "7:00pm" radio button is selected. The ticket count dropdown shows "3". The "Book Ticket" button is highlighted with a blue border. The bottom section of the window now displays the following text:

Movie Name: Mother India  
Show Time: 7:00PM  
You selected 3 ticket(s)  
Total Cost: ₹300

**Practical No. 26:** 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.

**Source Code:**

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.sql.*;

class MyFrame extends JFrame implements ActionListener
{
    JTextField name , code, designation , salary;

    MyFrame()
    {
        this.setSize(300,400);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setLayout(new BorderLayout());
        JPanel panel = new JPanel(new GridLayout(6,3,10,10));
        panel.setBorder(BorderFactory.createEmptyBorder(20,20,20,20));
        name = new JTextField();
        code = new JTextField();
        designation = new JTextField();
        salary = new JTextField();
        panel.add(new JLabel("NAME")); panel.add(name);
        panel.add(new JLabel("CODE")); panel.add(code);
        panel.add(new JLabel("DESIGNATION")); panel.add(designation);
        panel.add(new JLabel("SALARY")); panel.add(salary);
        JButton save = new JButton("SAVE");
        JButton reset = new JButton("RESET");
        JButton exit = new JButton("EXIT");
```

```

        save.addActionListener(this);
        reset.addActionListener(this);
        exit.addActionListener(this);
        panel.add(new JLabel());
        panel.add(save);
        panel.add(reset);
        panel.add(exit);
        this.add(panel , BorderLayout.NORTH);
        this.setVisible(true);
    }

    public void saveOperation()
    {
        try
        {
            Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/db" ,
"root" , "");

            String query = "INSERT INTO employee (`Name` , `Code` , `Designation` ,
`Salary`) VALUES (?, ?, ?, ?)";

            PreparedStatement st = con.prepareStatement(query);

            String vName = name.getText();
            int vCode = Integer.parseInt(code.getText());
            String vDesignation = designation.getText();
            float vSalary = Float.parseFloat(salary.getText());

            st.setString(1,vName);
            st.setInt(2,vCode);
            st.setString(3,vDesignation);
            st.setFloat(4,vSalary);

            int row = st.executeUpdate();

            if(row > 0)

                System.out.println("Saved data successfully.");
        }
    }
}

```

```

        else

            System.out.println("Data Could'nt be saved.");

            st.close();

            con.close();

        }catch(SQLException e)

        {

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

        }

    }

    public void actionPerformed(ActionEvent e)

    {

        String command = e.getActionCommand();

        if(command.equals("SAVE"))

        {

            saveOperation();

        }else if(command.equals("RESET"))

        {

            name.setText("");

            code.setText("");

            salary.setText("");

            designation.setText("");

        }else if(command.equals("EXIT"))

        {

            System.exit(0);

        }

    }

}

public class Q26 {

    public static void retrieveAllRecords()

```

```

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

        String query = "SELECT * FROM employee";

        PreparedStatement st = con.prepareStatement(query);

        ResultSet rs = st.executeQuery();

        int i = 1;

        while(rs.next())

        {

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

            System.out.println(rs.getString(1) + " " + rs.getInt(2) + " " + rs.getString(3) + " "
+ rs.getFloat(4));

        }

        rs.close();

        st.close();

        con.close();

    }catch(SQLException e)

    {

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

    }

}

public static void main(String[] args) {

    new MyFrame();

    retrieveAllRecords();

}

}

```



## Output:

A Java Swing window titled 'Employee Insert' with a light gray background. It contains four text input fields for 'NAME', 'CODE', 'DESIGNATION', and 'SALARY'. The 'NAME' field is filled with 'Himanshu Raturi', 'CODE' with '20', 'DESIGNATION' with 'SWE', and 'SALARY' with '50000'. Below the fields are three buttons: 'SAVE', 'RESET', and 'EXIT'.

A screenshot of the phpMyAdmin web interface. The left sidebar shows a database structure with 'db' selected and 'employee' as the current table. The main area displays the table data with columns: Name, Code, Designation, and Salary. The table contains 6 rows of data.

Name	Code	Designation	Salary
Himanshu Raturi	20	SWE	50000
Bhaumik Negi	15	CyberSecurity Expert	45000
Ishant Chamoli	10	Cloud Engineer	30000
Priyanshu Joshi	25	AI/ML engineer	60000
Tanish Kothiyal	5	Solution Architect	45000
Admin 2	41	Cloud Architect	40000

```
PS C:\Users\Himanshu\Desktop\Coding\CODS\Java\Himanshu_Raturi> javac -cp ".;mysql-connector-j-9.3.0.jar" Q26.java
PS C:\Users\Himanshu\Desktop\Coding\CODS\Java\Himanshu_Raturi> java -cp ".;mysql-connector-j-9.3.0.jar" Q26
Employee 1:Himanshu Raturi    20  SWE  50000.0
Employee 2:Bhaumik Negi      15  CyberSecurity Expert  45000.0
Employee 3:Ishant Chamoli    10  Cloud Engineer  30000.0
Employee 4:Priyanshu Joshi   25  AI/ML engineer  60000.0
Employee 5:Tanish Kothiyal   5   Solution Architect  45000.0
Employee 6:Admin 2          41  Cloud Architect  40000.0
PS C:\Users\Himanshu\Desktop\Coding\CODS\Java\Himanshu_Raturi>
```