

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:

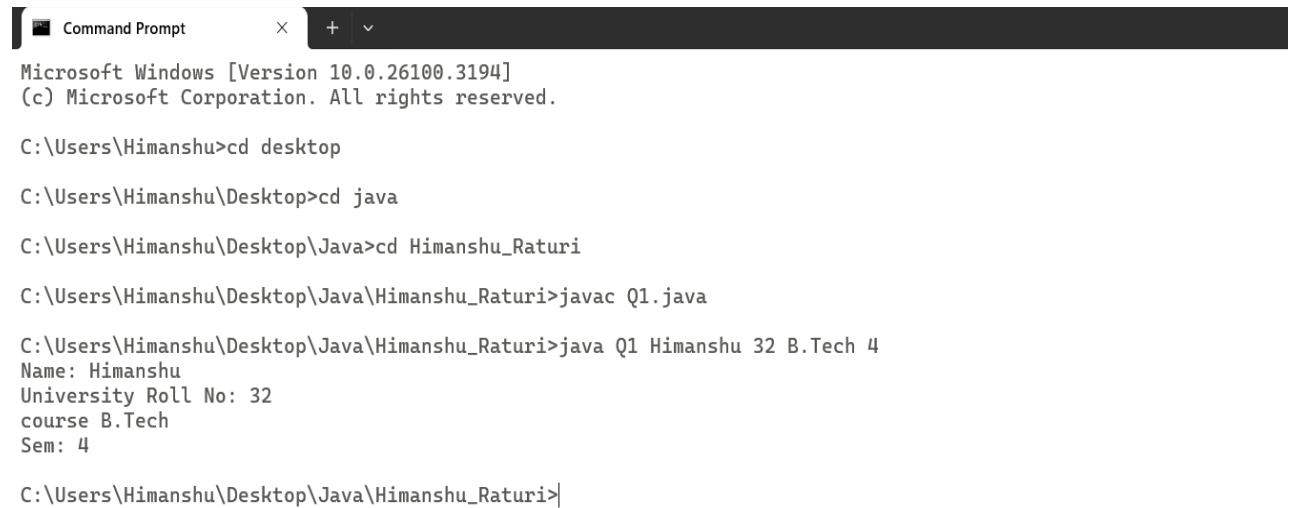
Course:

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.*;

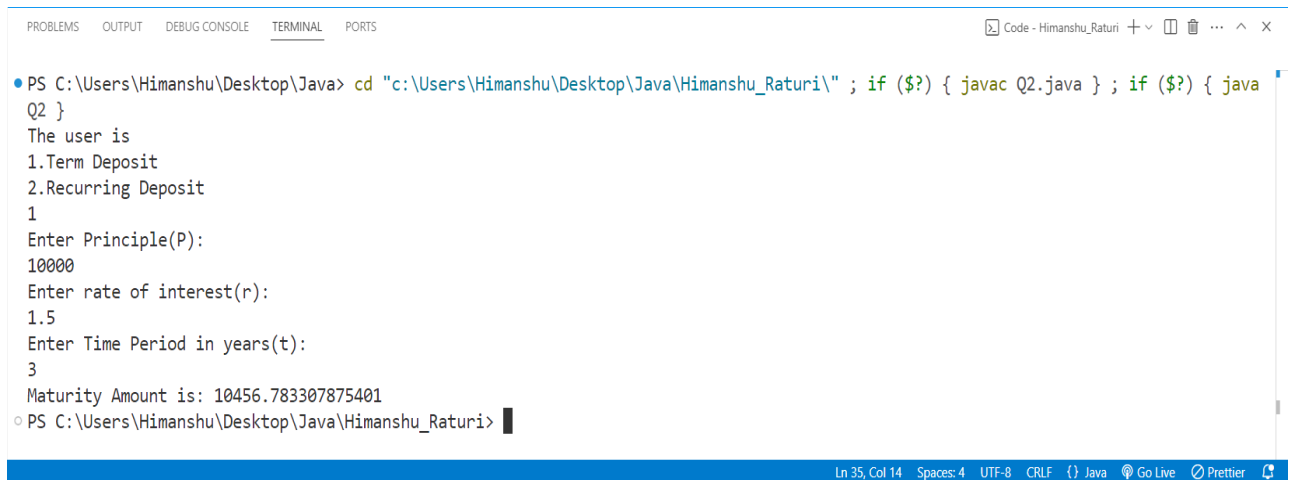
public class Q2
{
    public static void main(String args[])
    {
        System.out.println("The user is\n1.Term Deposit\n2.Recurring Deposit");
        Scanner sc = new Scanner(System.in);
        int Choice = sc.nextInt();
        switch(Choice)
        {
            case 1:
            {
                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;

            }

        }

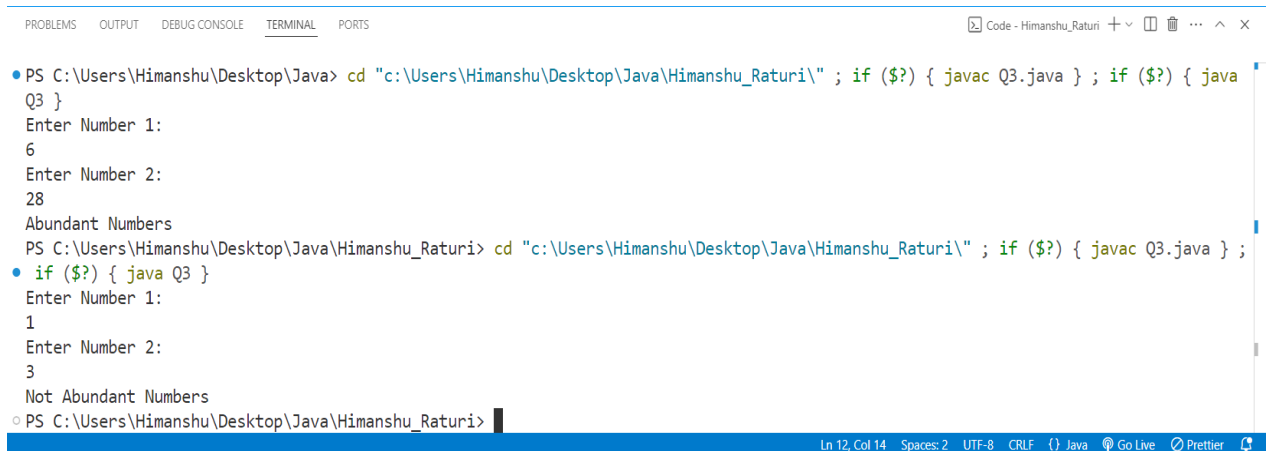
        if ((sum1 == num1) && (sum2 == num2)) {

            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:

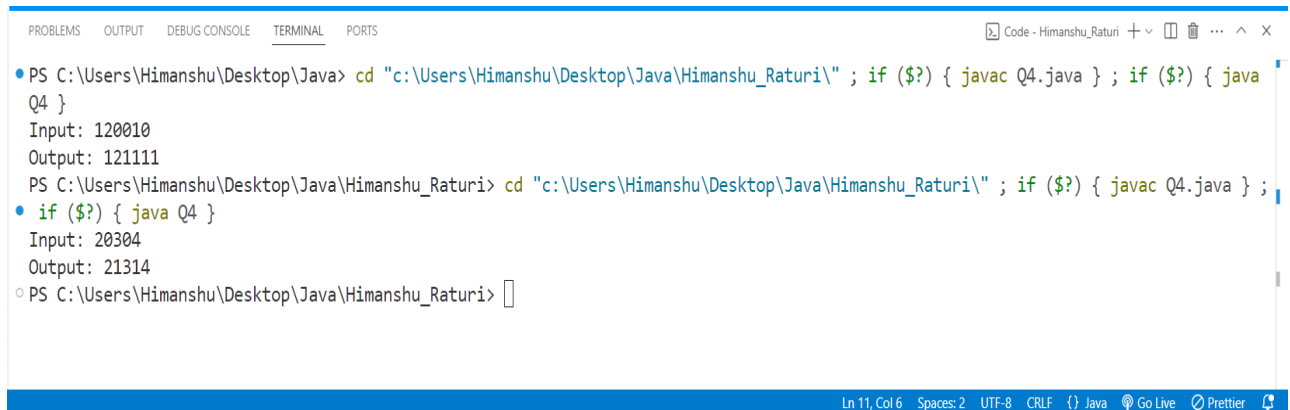
```
import java.util.Scanner;

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];

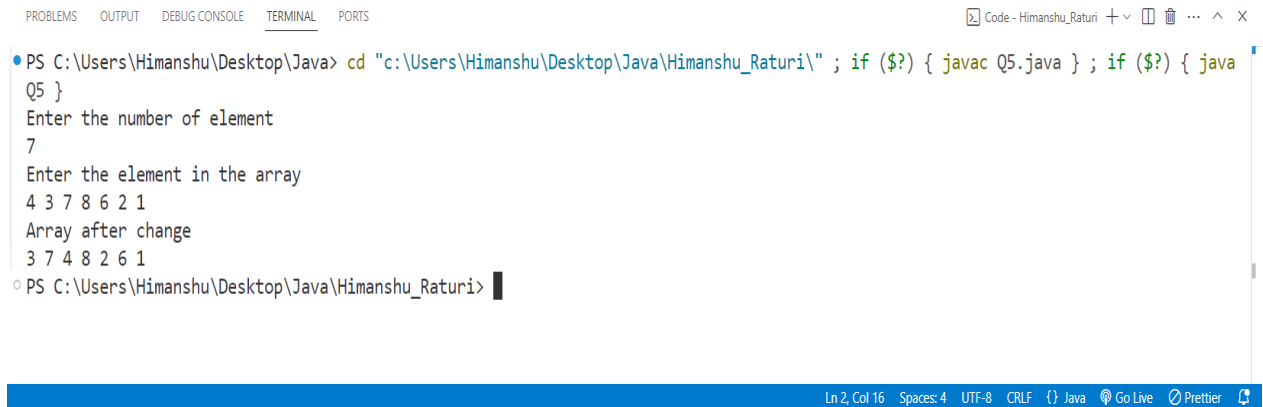
                        arr[i]=arr[i+1];
```

```

        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:



```
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>
```

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];

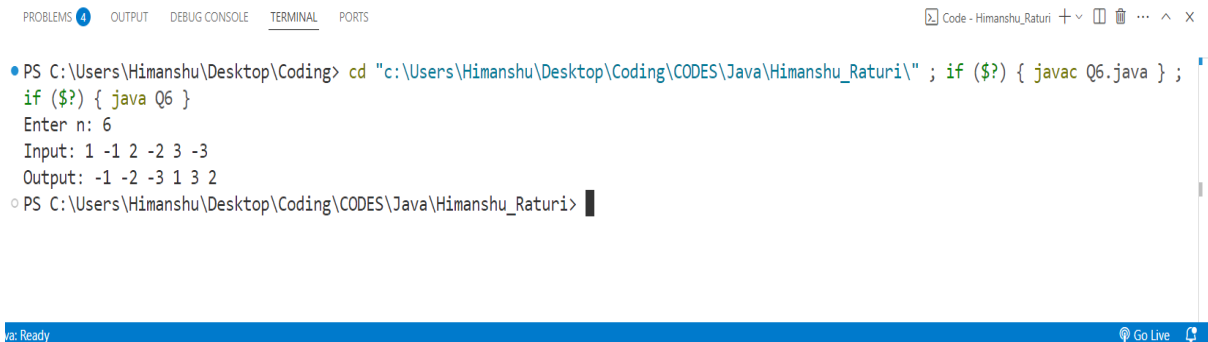
                arr[i] = arr[first];

                arr[first] = temp;

                first++;
            }
        }
    }
}
```

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


Output:



The screenshot shows a VS Code interface with a terminal window open. The terminal has tabs for PROBLEMS (4), OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The active tab is TERMINAL, showing a PowerShell prompt at C:\Users\Himanshu\Desktop\Coding\CODES\Java\Himanshu_Raturi. The user has entered a command to compile and run a Java file. The output shows the input sequence and the resulting output sequence.

```
PS C:\Users\Himanshu\Desktop\Coding\CODES\Java\Himanshu_Raturi> 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>
```

At the bottom of the terminal window, there is a status bar that says "Ready" and a "Go Live" button.

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;

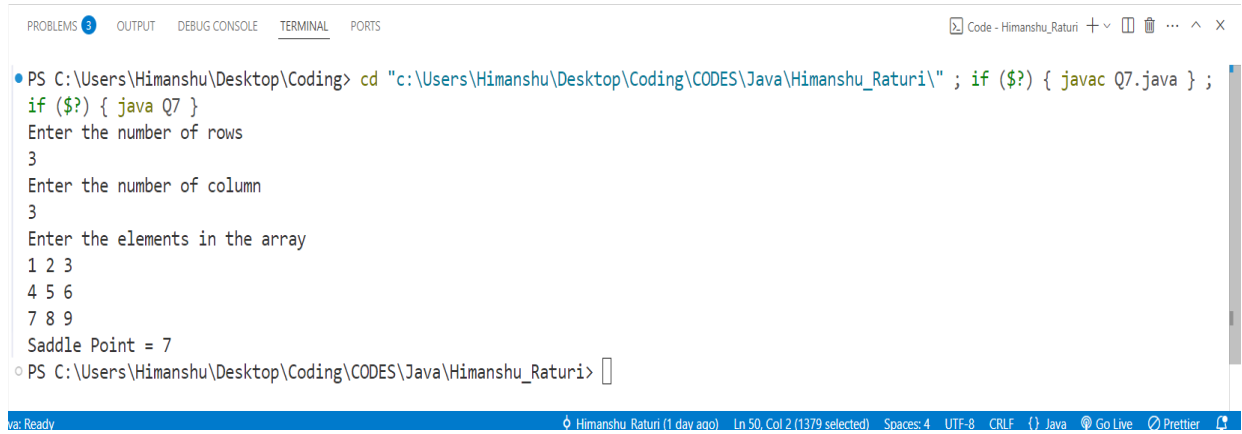
            for(int j=0;j<columns;j++)
            {
```

```

        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:

```
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 ($?) { javac Q9_Bank.java
} ; if ($?) { 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
```

```
PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL PORTS Code - Himanshu_Raturi + - [ ] ... ^ x
2 to withdraw money
3 to Change address
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 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
```

```
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>
```

VS Code status bar: 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 + " ");
    }
}

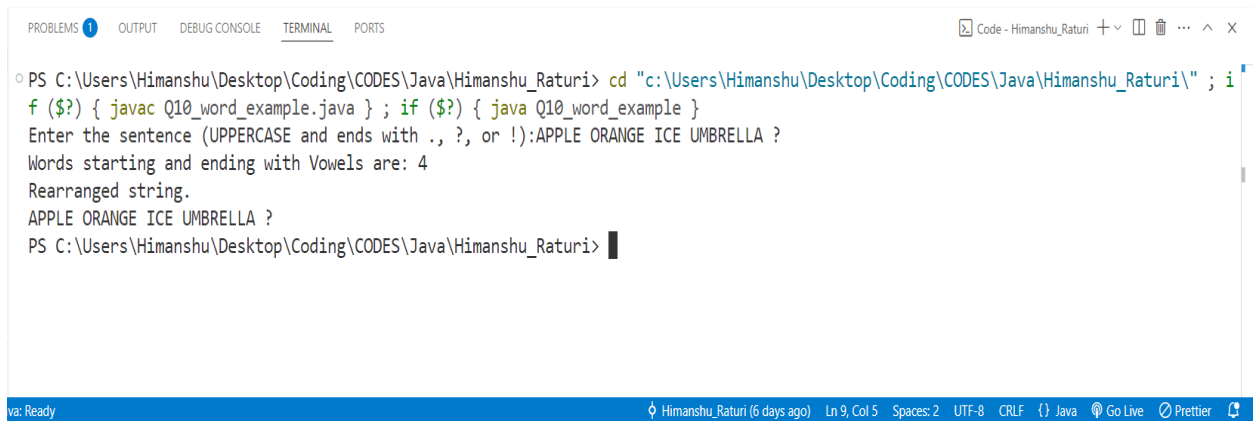
System.out.println("Rearranged string.");
System.out.print(vowelWords);
System.out.println(otherWords + " ");

}

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++] = A[i++];
}
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];
    }

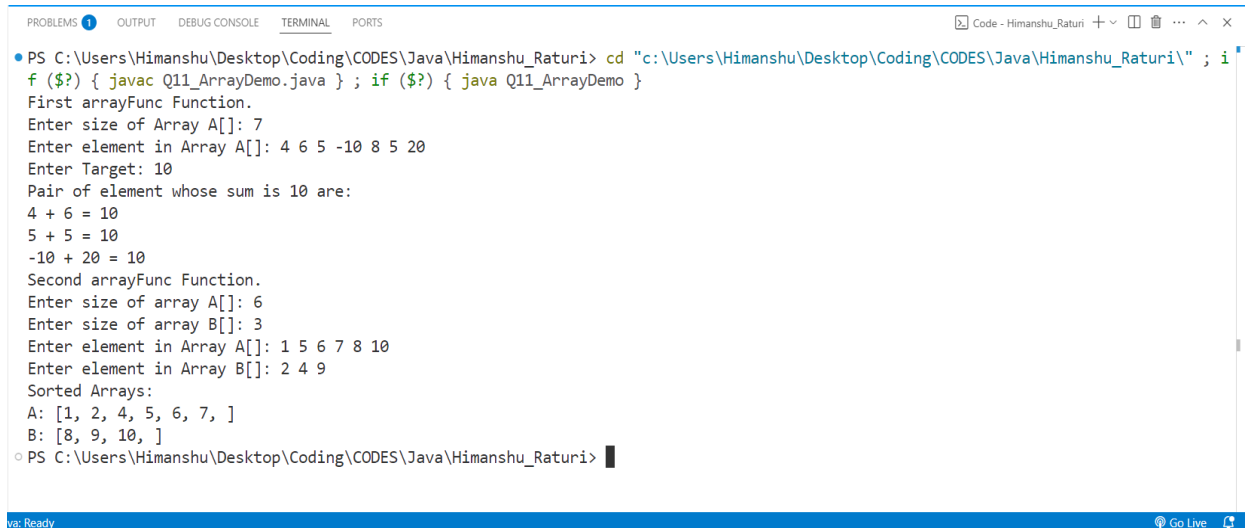
```

```

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.

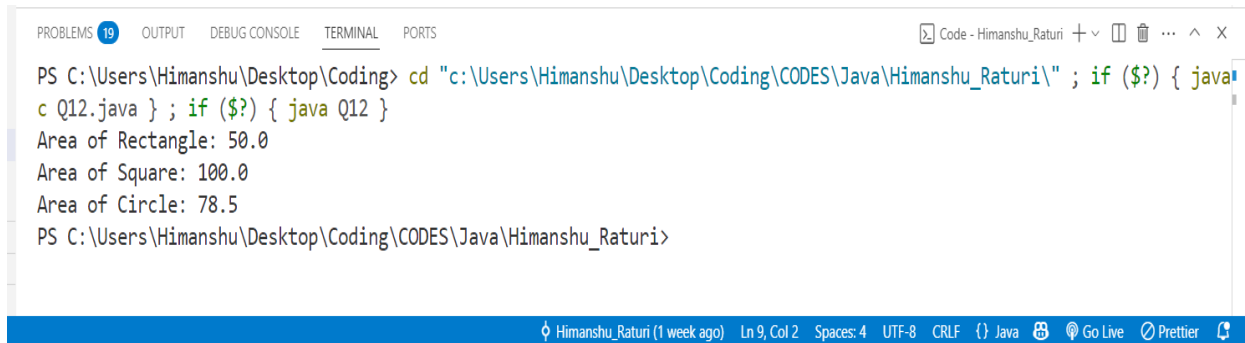
Source Code:

```
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);
    }
}

```

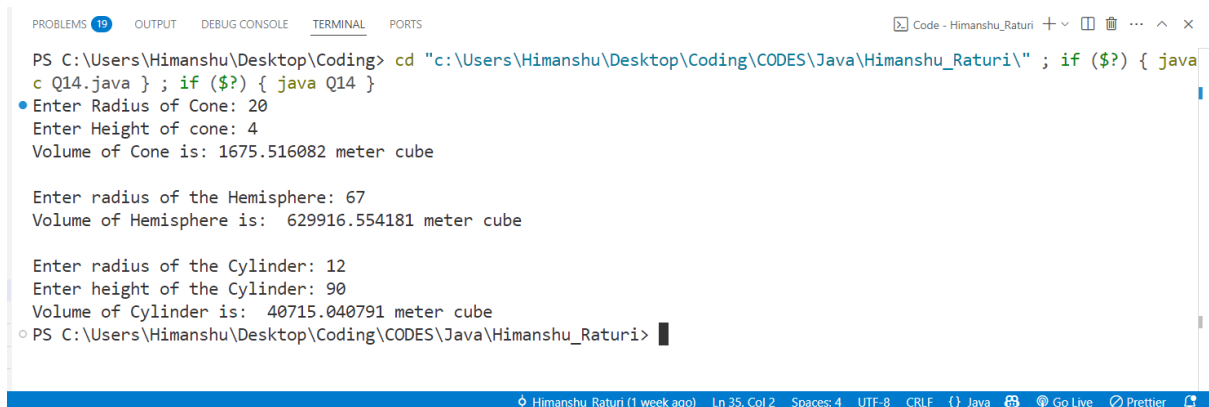
```
System.out.print("Enter Radius of Cone: ");  
double cradius = sc.nextDouble();  
System.out.print("Enter Height of cone: ");  
double cheight = sc.nextDouble();  
Shape cone = new Cone(cradius, cheight);  
cone.volume();
```

```
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();
    }
}

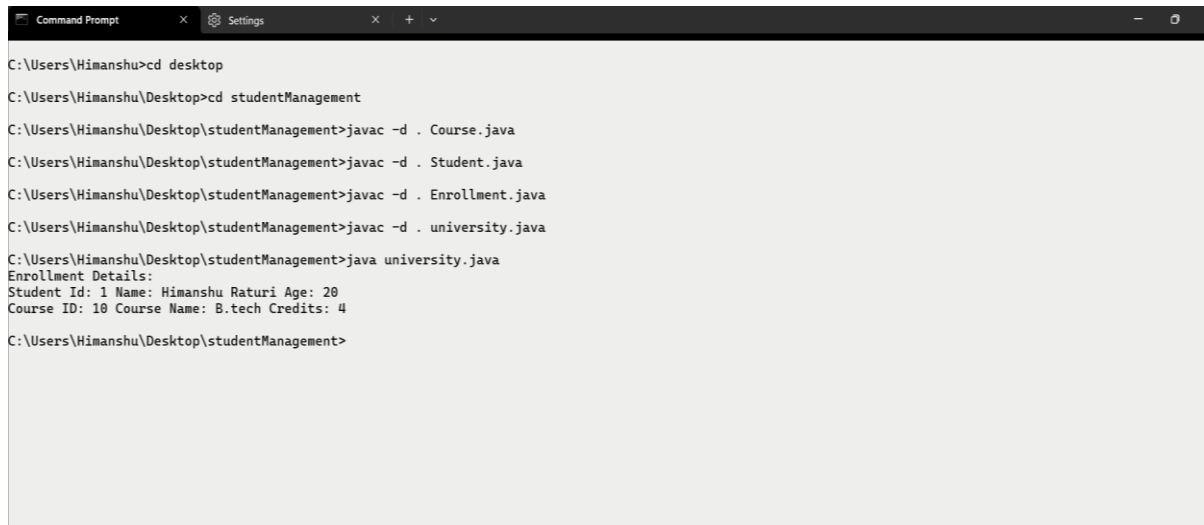
default package

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>
```