ASSIGNMENT DAY 4

CSA0976 JavaProgramming

Name: K.KALYAN

Reg No: 192110258

**QUESTION 1.**

import java.io.BufferedReader; import java.io.FileReader; import java.io.IOException; public class CountWordsCharsLines {

public static void main(String[] args) { String fileName = "File1.txt"; int wordCount = 0; int charCount = 0; int lineCount = 0;

try (BufferedReader br = new BufferedReader(new FileReader(fileName))) {

String line;

while ((line = br.readLine()) != null) { lineCount++;

String[] words = line.split("\\s+"); wordCount += words.length; for (String word : words) { charCount += word.length();

}

}

} catch (IOException e) {

e.printStackTrace();

}

System.out.println("Number of words: " + wordCount);

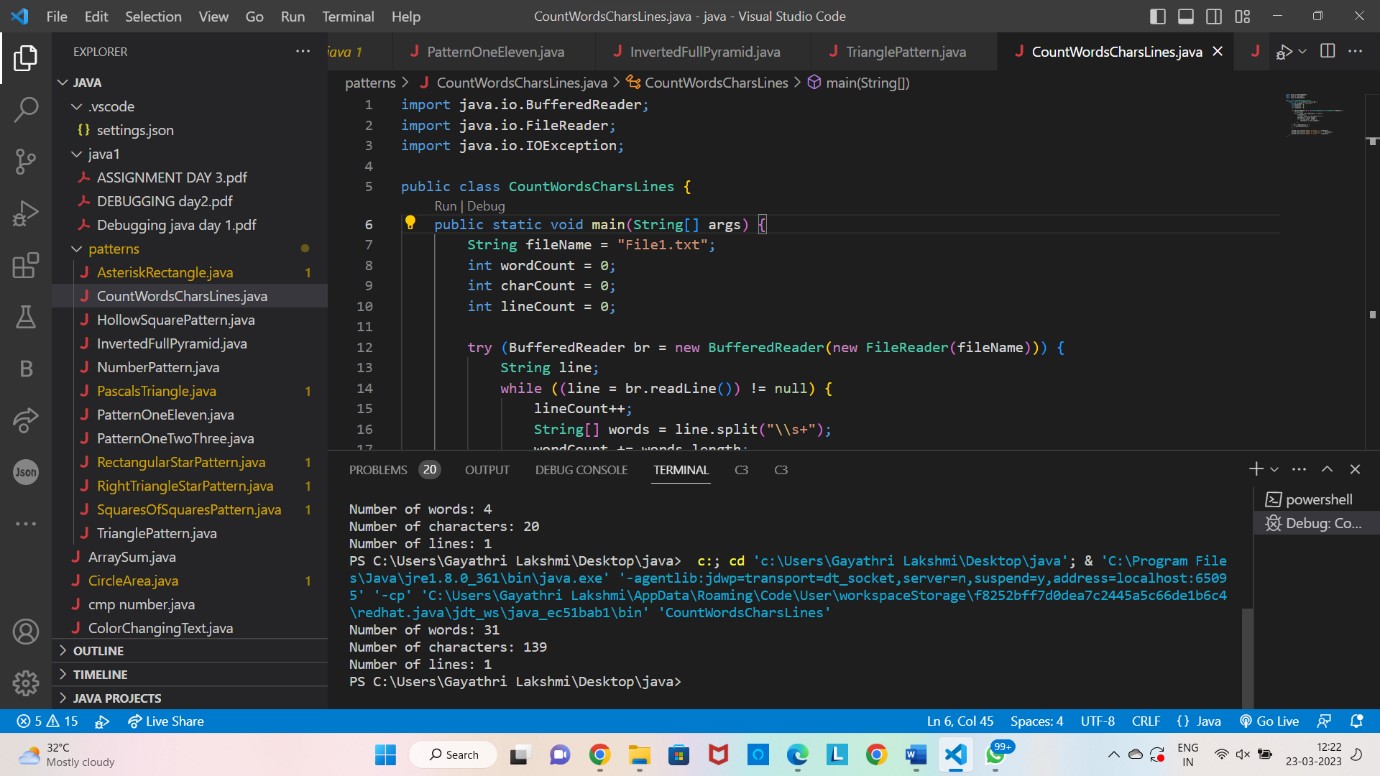
System.out.println("Number of characters: " + charCount);

System.out.println("Number of lines: " + lineCount);

}

}

OUTPUT:



**QUESTION 2.**

import java.util.Scanner; public class Customer { private int accountNo; private String accName; private double balance; public Customer(int accountNo, String accName, double balance) { this.accountNo = accountNo; this.accName = accName; this.balance = balance;

}

public synchronized void deposit(double amount) { System.out.println("Depositing " + amount + "..."); balance += amount;

System.out.println("Deposit complete. New balance: " + balance);

notify();

}

public synchronized void withdraw(double amount) { while (balance < amount) {

System.out.println("Insufficient balance. Waiting for deposit...");

try { wait();

} catch (InterruptedException e) {

e.printStackTrace();

}

}

System.out.println("Withdrawing " + amount + "..."); balance -= amount;

System.out.println("Withdrawal complete. New balance: " + balance);

}

public static void main(String[] args) {

Customer customer = new Customer(12345, "John Doe", 100.0);

Scanner scanner = new Scanner(System.in); System.out.print("Enter withdrawal amount: "); double withdrawAmount = scanner.nextDouble();

Thread withdrawThread = new Thread(() -> customer.withdraw(withdrawAmount));

withdrawThread.start();

try {

Thread.sleep(1000);

} catch (InterruptedException e) {

e.printStackTrace();

}

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

double depositAmount = scanner.nextDouble();

Thread depositThread = new Thread(() -> customer.deposit(depositAmount));

depositThread.start();

try {

withdrawThread.join(); depositThread.join();

} catch (InterruptedException e) {

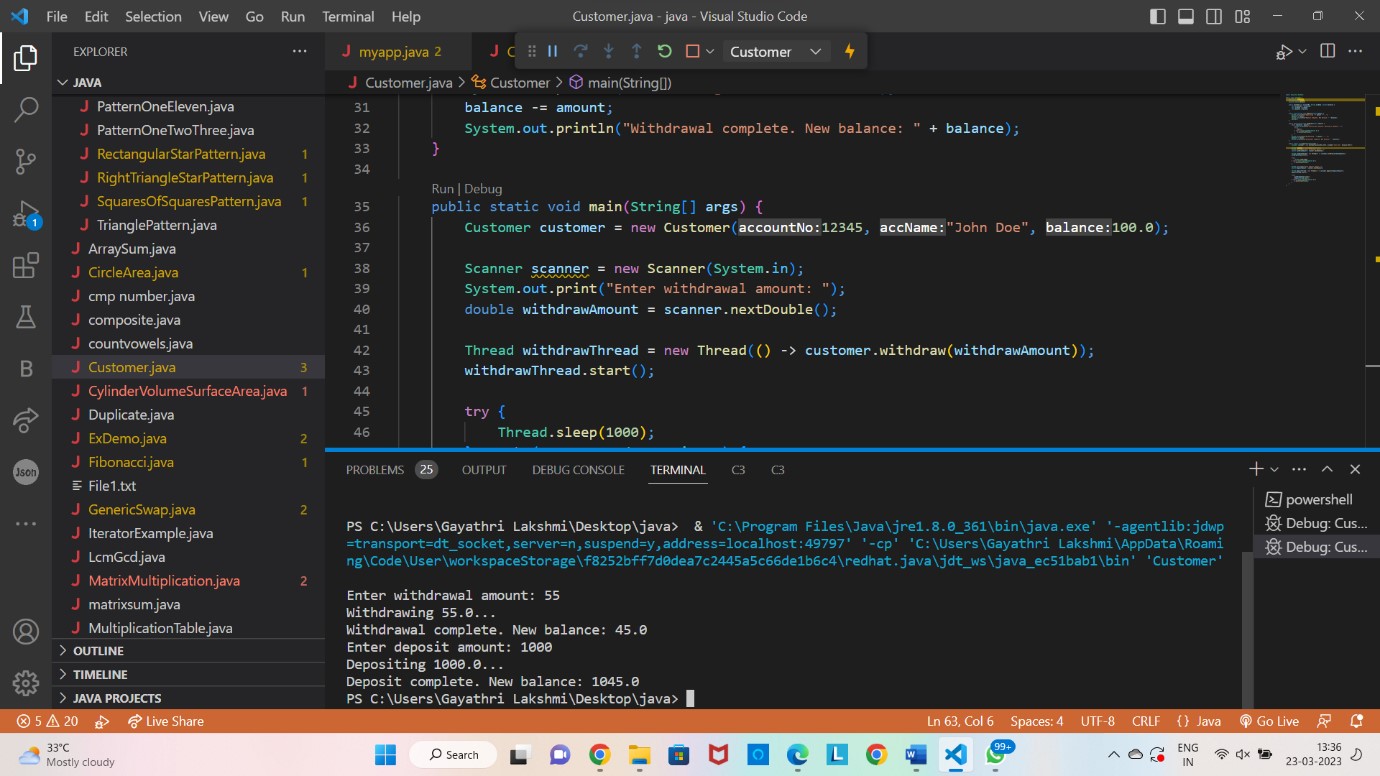
e.printStackTrace();

}

}

}

OUTPUT:



**QUESTION 3.**

import java.util.\*;

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

Scanner input = new Scanner(System.in); System.out.print("n: "); int n = input.nextInt();

List<String> answer = fizzBuzz(n);

System.out.println(answer);

}

public static List<String> fizzBuzz(int n) { List<String> answer = new ArrayList<String>();

for (int i = 1; i <= n; i++) { if (i % 3 == 0 && i % 5 == 0) { answer.add("FizzBuzz"); } else if (i % 3 == 0) { answer.add("Fizz"); } else if (i % 5 == 0) { answer.add("Buzz");

} else {

answer.add(Integer.toString(i));

}

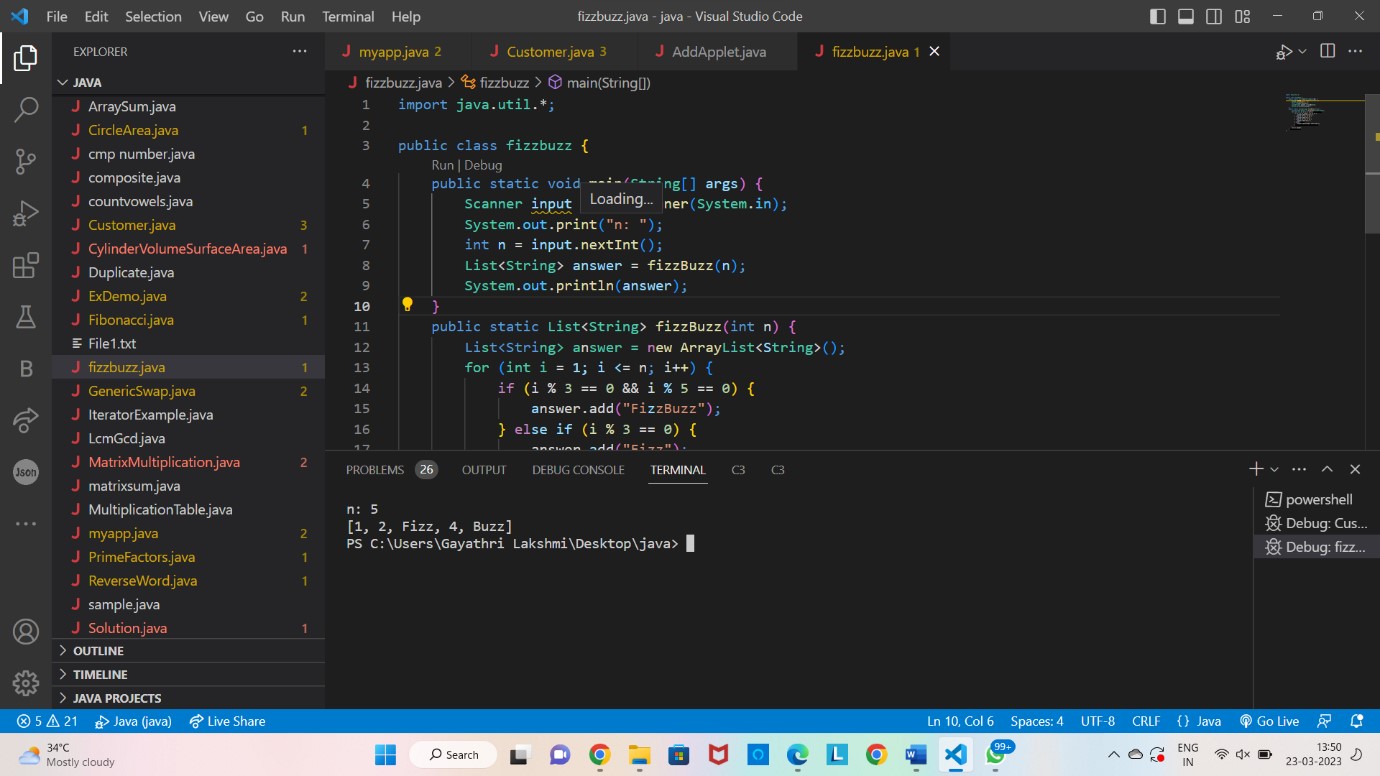
}

return answer;

}

}

OUTPUT:



QUESTION 4.

import java.util.\*; class Solution1 { public static void main(String[] args) {

Scanner input = new Scanner(System.in);

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

String s = input.nextLine();

System.out.print("goal: "); String goal = input.nextLine(); boolean result = rotateString(s, goal);

System.out.println(result);

}

public static boolean rotateString(String s, String goal) { if (s.length() != goal.length()) { return false;

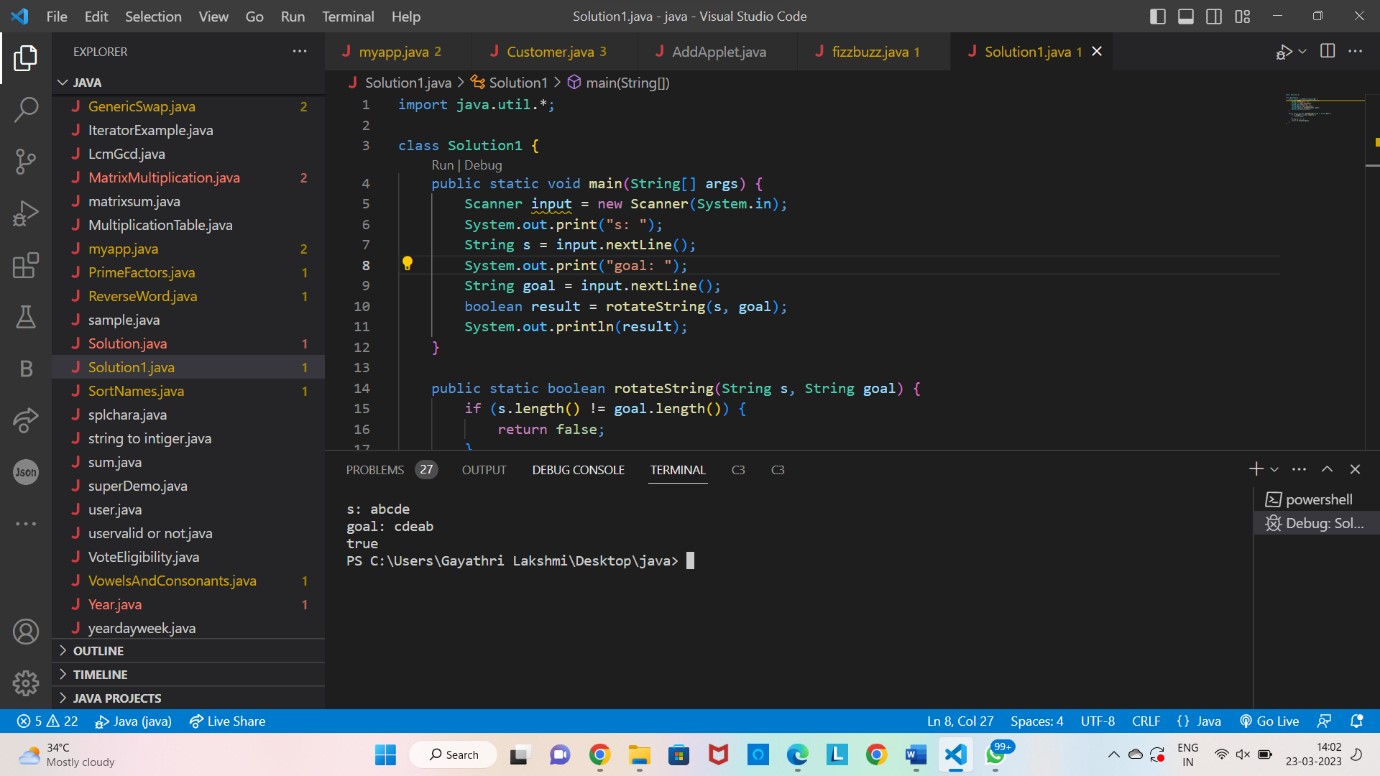
}

String s2 = s + s; return s2.contains(goal);

}

}

OUTPUT:



**QUESTION 5 DEBUGGING:**

class PrimeExample implements Runnable {

public void run() { int i, m = 20, flag; for (i = 1; i <= m; i++) { flag = 1; if (i <= 3) { System.out.println(i + " is prime number");

continue; } else { for (int j = 2; j < i; j++) { if (i % j == 0) { flag = 0; break;

} } if (flag != 1) {

System.out.println(i + " is not prime number");

} else {

System.out.println(i + " is prime number");

}

}

}

}

}

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

try {

PrimeExample p1 = new PrimeExample(); Thread t1 = new Thread(p1);

t1.start();

} catch (Exception e) {

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

}

}

}

**OUTPUT:**

