

ASSIGNMENT DAY 4

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);

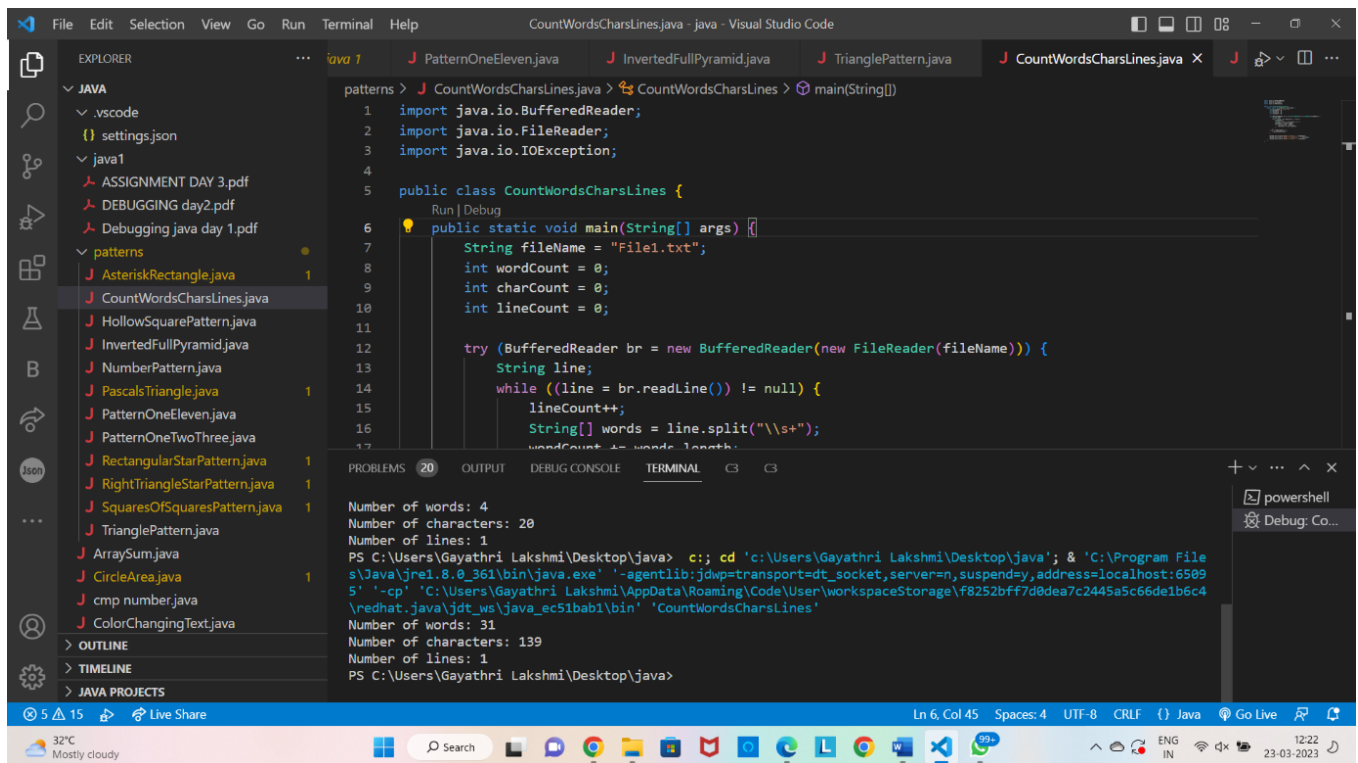
        System.out.println("Number of characters: " + charCount);

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

    }

}
```

OUTPUT:



```
patterns > J CountWordsCharsLines.java > CountWordsCharsLines > main(String[])
1  import java.io.BufferedReader;
2  import java.io.FileReader;
3  import java.io.IOException;
4
5  public class CountWordsCharsLines {
6      public static void main(String[] args) {
7          String fileName = "File1.txt";
8          int wordCount = 0;
9          int charCount = 0;
10         int lineCount = 0;
11
12         try (BufferedReader br = new BufferedReader(new FileReader(fileName))) {
13             String line;
14             while ((line = br.readLine()) != null) {
15                 lineCount++;
16                 String[] words = line.split("\\s+");
17                 wordCount += words.length;
18                 charCount += line.length();
19             }
20         } catch (IOException e) {
21             e.printStackTrace();
22         }
23
24         System.out.println("Number of words: " + wordCount);
25         System.out.println("Number of characters: " + charCount);
26         System.out.println("Number of lines: " + lineCount);
27     }
28 }
```

Number of words: 4
Number of characters: 20
Number of lines: 1

PS C:\Users\Gayathri Lakshmi\Desktop\java> c:: cd 'c:\Users\Gayathri Lakshmi\Desktop\java'; & 'C:\Program File
s\Java\jre1.8.0_361\bin\java.exe' -agentlib:jdwp=transport=dt_socket,server=n,suspend=y,address=localhost:6509
5' -cp 'C:\Users\Gayathri Lakshmi\AppData\Roaming\Code\User\workspaceStorage\f8252bff7d0dea7c2445a5c66de1b6c4
\redhat.java\jdt_ws\java_ec51bab1\bin' 'CountWordsCharsLines'
Number of words: 31
Number of characters: 139
Number of lines: 1
PS C:\Users\Gayathri Lakshmi\Desktop\java>

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) {
```

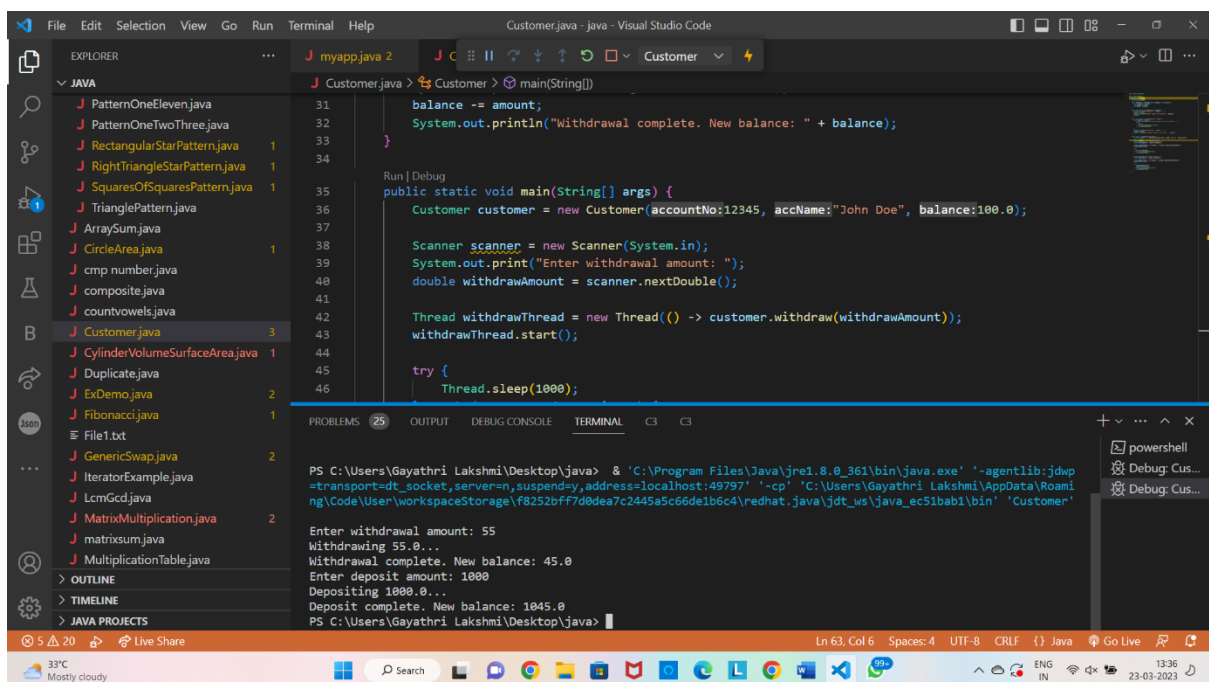
```
    e.printStackTrace();
```

```
}
```

```
}
```

```
}
```

OUTPUT:

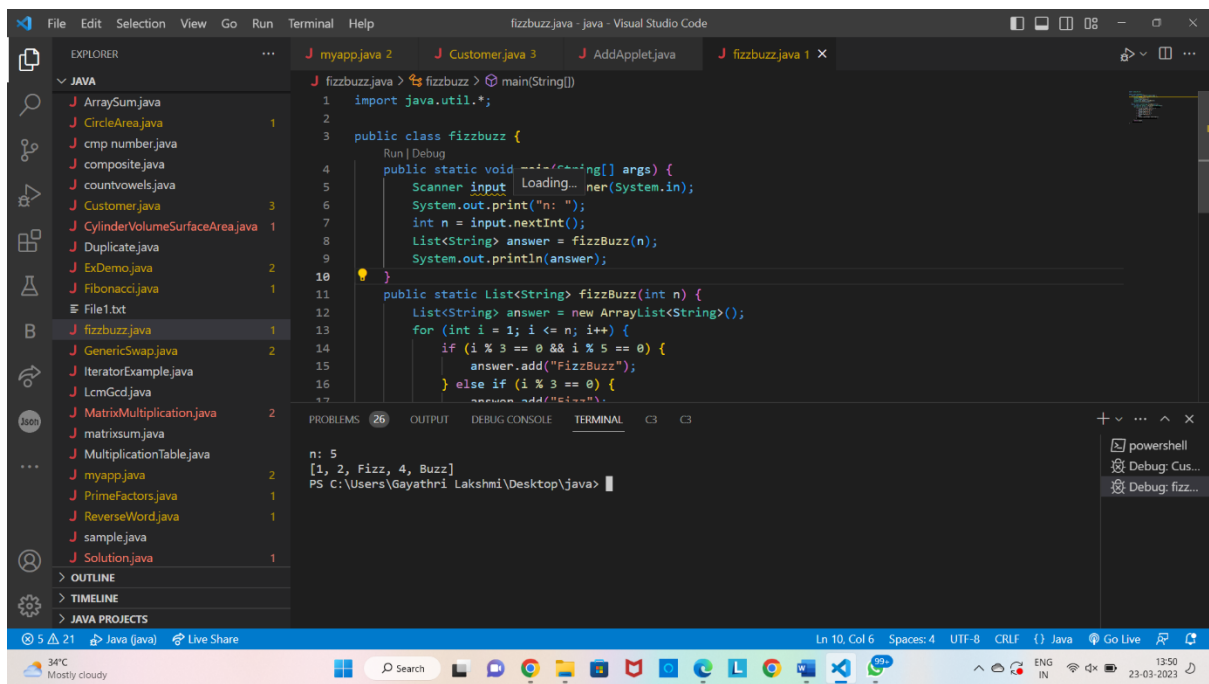


QUESTION 3.

```
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;

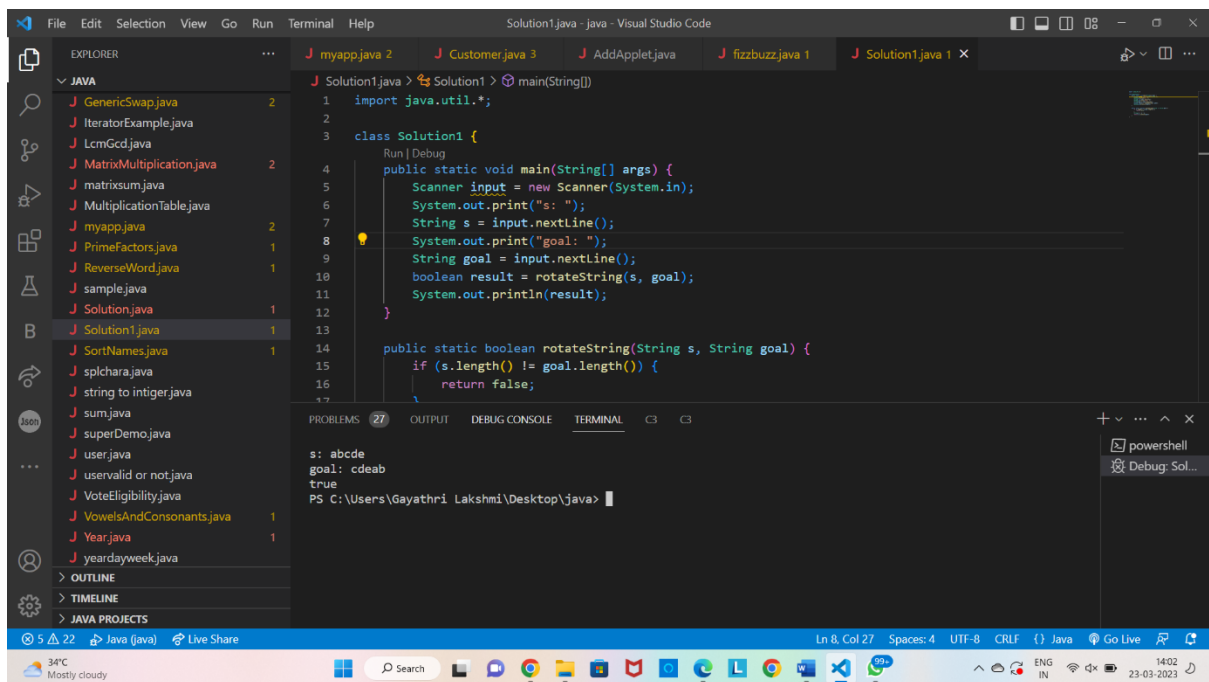
return s2.contains(goal);

}

}

```

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) {

```


OUTPUT:

The screenshot displays the Visual Studio Code interface with the following components:

- EXPLORER:** A list of Java files including ExDemo.java, Fibonacci.java, File1.txt, fizzbuzz.java, GenericSwap.java, IteratorExample.java, LcmGcd.java, MatrixMultiplication.java, matrixsum.java, MultiplicationTable.java, myapp.java, PrimeExample.java, PrimeFactors.java, ReverseWord.java, sample.java, Solution.java, Solution1.java, SortNames.java, splchara.java, string to intiger.java, sum.java, superDemo.java, user.java, and others.
- EDITOR:** The file PrimeExample.java is open, showing the following code:

```
1 class PrimeExample implements Runnable {
2     public void run() {
3         int i, m = 20, flag;
4         for (i = 1; i <= m; i++) {
5             flag = 1;
6             if (i <= 3) {
7                 System.out.println(i + " is prime number");
8                 continue;
9             } else {
10                for (int j = 2; j < i; j++) {
11                    if (i % j == 0) {
12                        flag = 0;
```
- OUTPUT:** The output of the program is displayed in the bottom right pane:

```
1 is prime number
2 is prime number
10 is not prime number
11 is prime number
12 is not prime number
13 is prime number
14 is not prime number
15 is not prime number
16 is not prime number
17 is prime number
18 is not prime number
19 is prime number
20 is not prime number
```
- TERMINAL:** The terminal shows the command used to run the program:

```
PS C:\Users\Gayathri Lakshmi\Desktop\java> .\PrimeExample.java
```