ASSIGNMENT

B. LIKHITHA

Assignment 1

Title: Write Data to CSV File

Problem Statement:

Write student details (id, name, marks) into a CSV file named students.csv.

Conditions:

• Use FileWriter

• Data format:

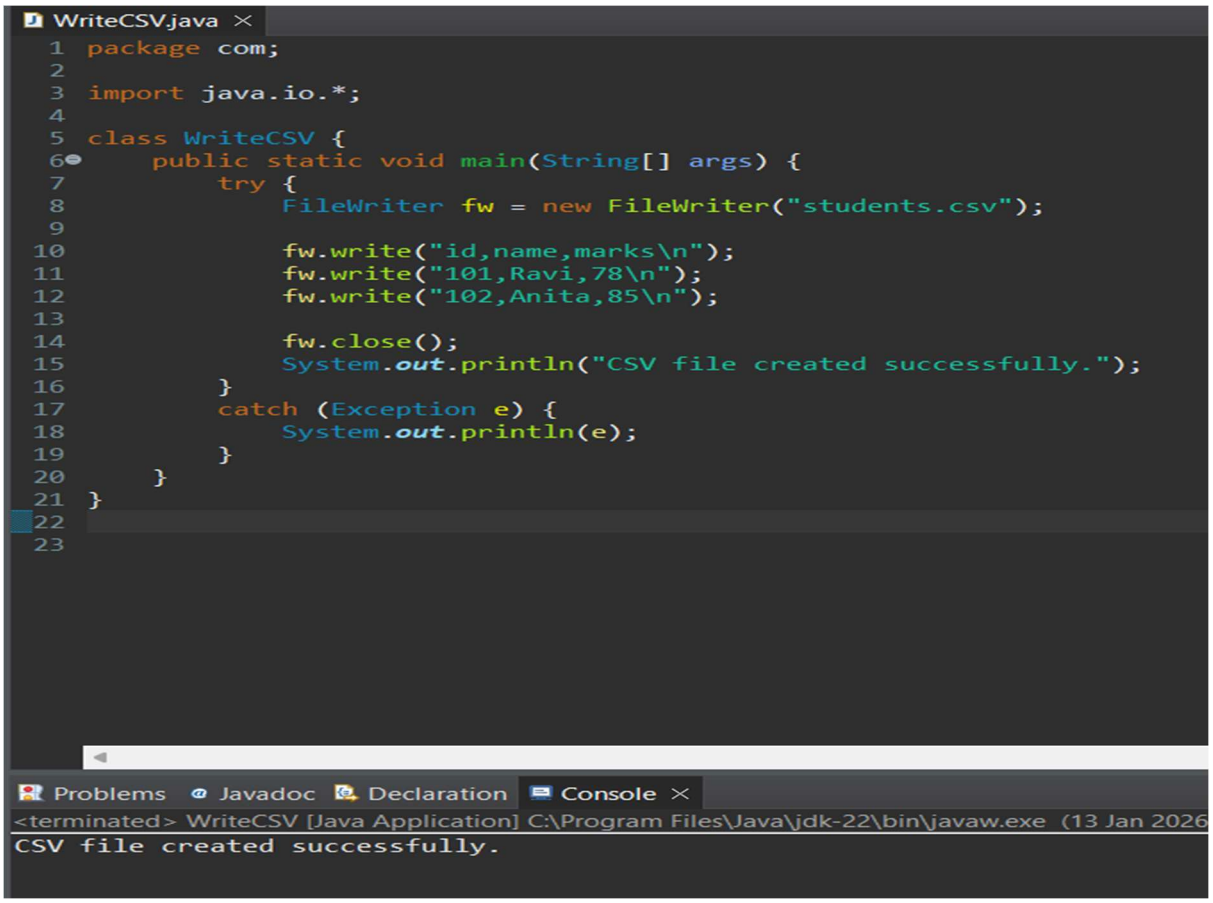
• id,name,marks

• 101,Ravi,78

• 102,Anita,85

Expected Outcome:

CSV file should be created and data should be stored correctly.



Assignment 2

Title: Read CSV File and Display Data

Problem Statement:

Read all records from students.csv and display them on the console.

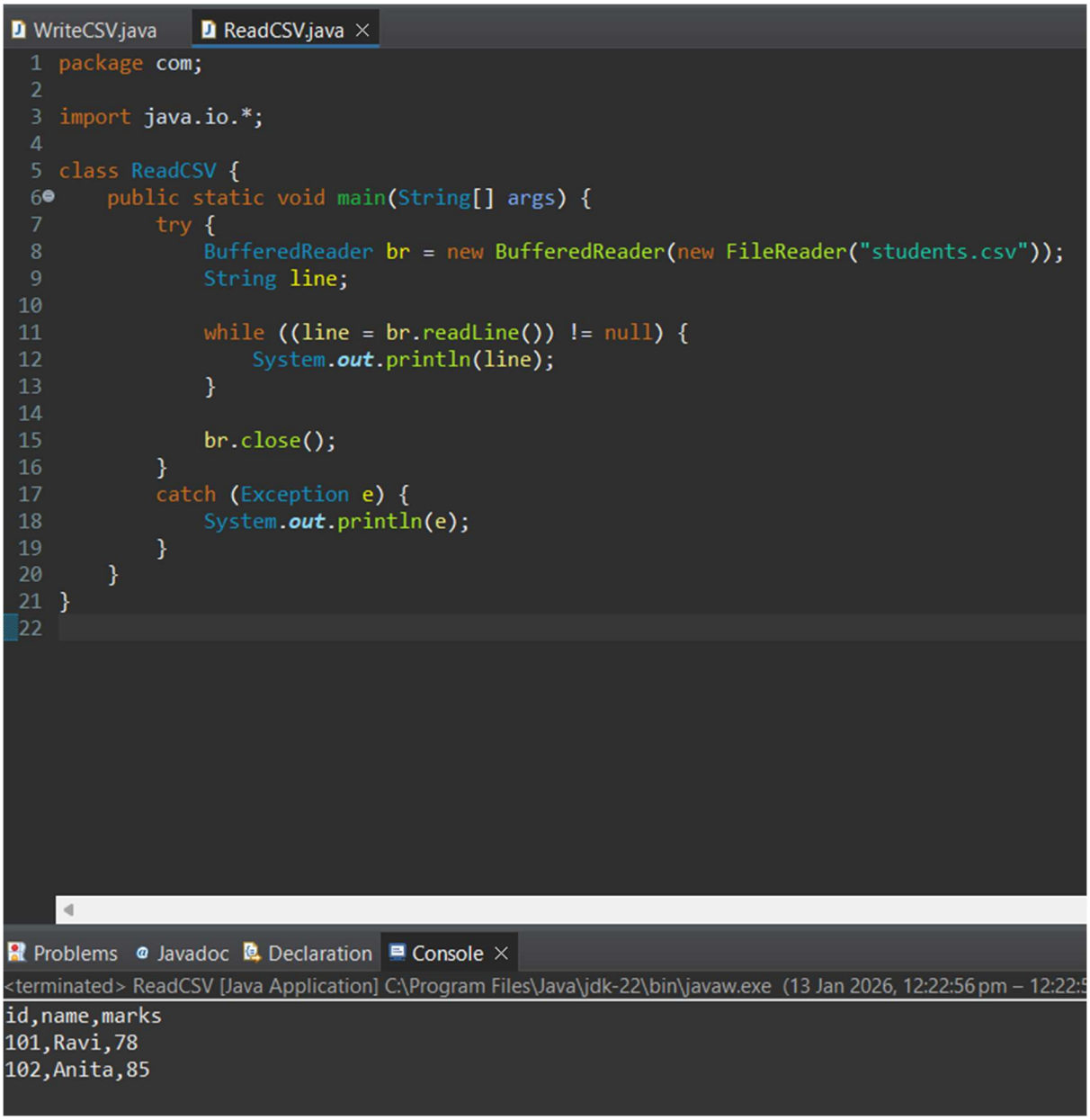
Conditions:

• Use FileReader and BufferedReader

• Split each line using comma (,)

Expected Outcome:

All rows should be printed.



Assignment 3

Title: Display Students with Marks Greater Than 60

Problem Statement:

Read students.csv and display only students whose marks are greater than 60.

Conditions:

• Convert marks to integer• Use if condition

Expected Outcome:

Only eligible students should be printed.

WriteCSV.java ReadCSV.java MarksAbove60.java

```java
package com;
import java.io.*;

class MarksAbove60 {
    public static void main(String[] args) {
        try {
            BufferedReader br = new BufferedReader(new FileReader("students.csv"));
            String line;

            br.readLine(); // skip header

            while ((line = br.readLine()) != null) {
                String[] data = line.split(",");
                int marks = Integer.parseInt(data[2]);

                if (marks > 60) {
                    System.out.println(line);
                }
            }
            br.close();
        }
        catch (Exception e) {
            System.out.println(e);
        }
    }
}
```

Problems Javadoc Declaration Console

```
<terminated> MarksAbove60 [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe (13 Jan 2026, 12:23:50 pm – 12:2
101,Ravi,78
102,Anita,85
```

Assignment 4

Title: Count Number of Records in CSV File

Problem Statement:

Count how many student records are present in students.csv.

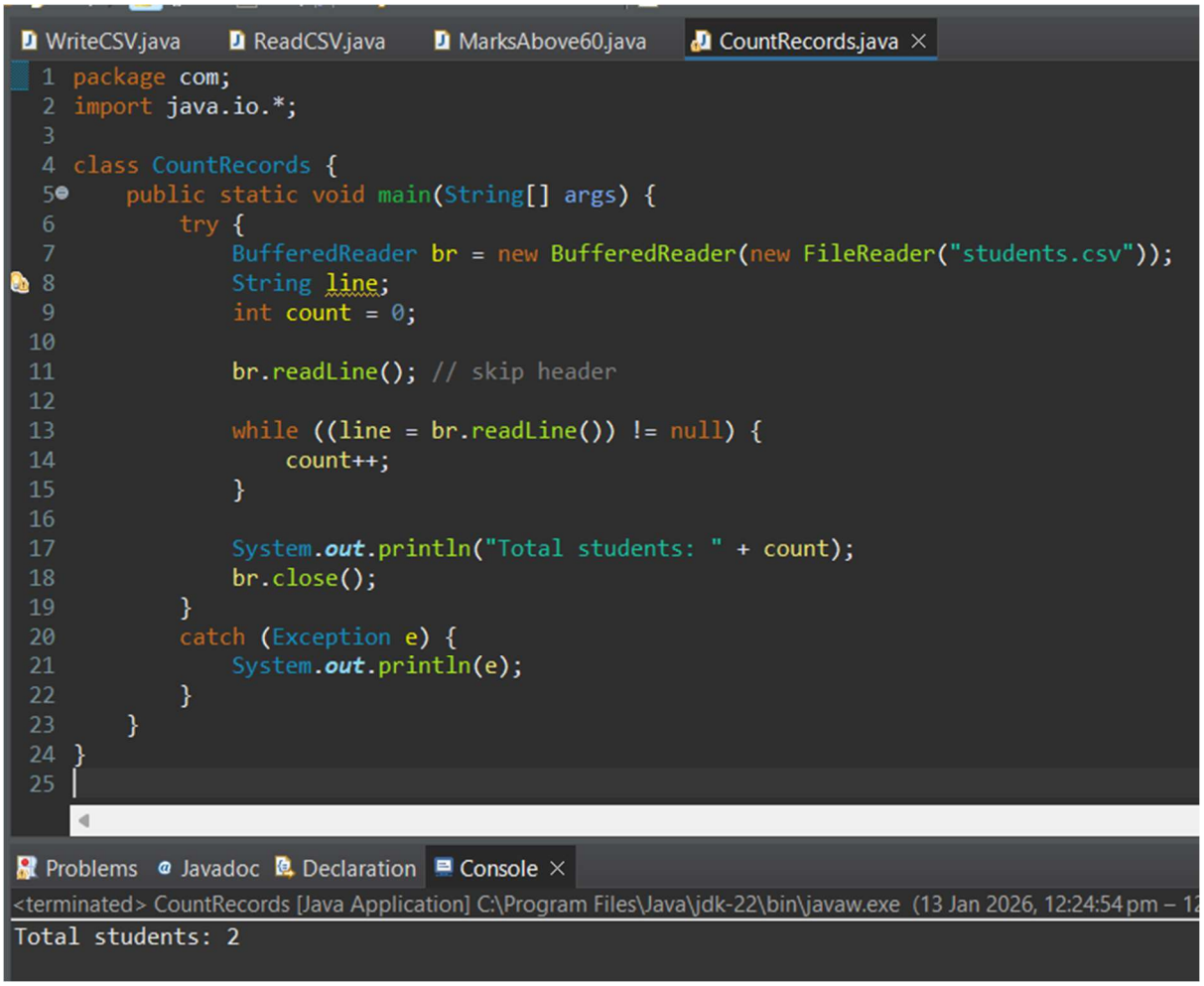
Conditions:

• Ignore header line

• Use counter variable

Expected Outcome:

Total number of students displayed.



Assignment 5

Title: Search Student by Name

Problem Statement:

Search a student by name from students.csv.

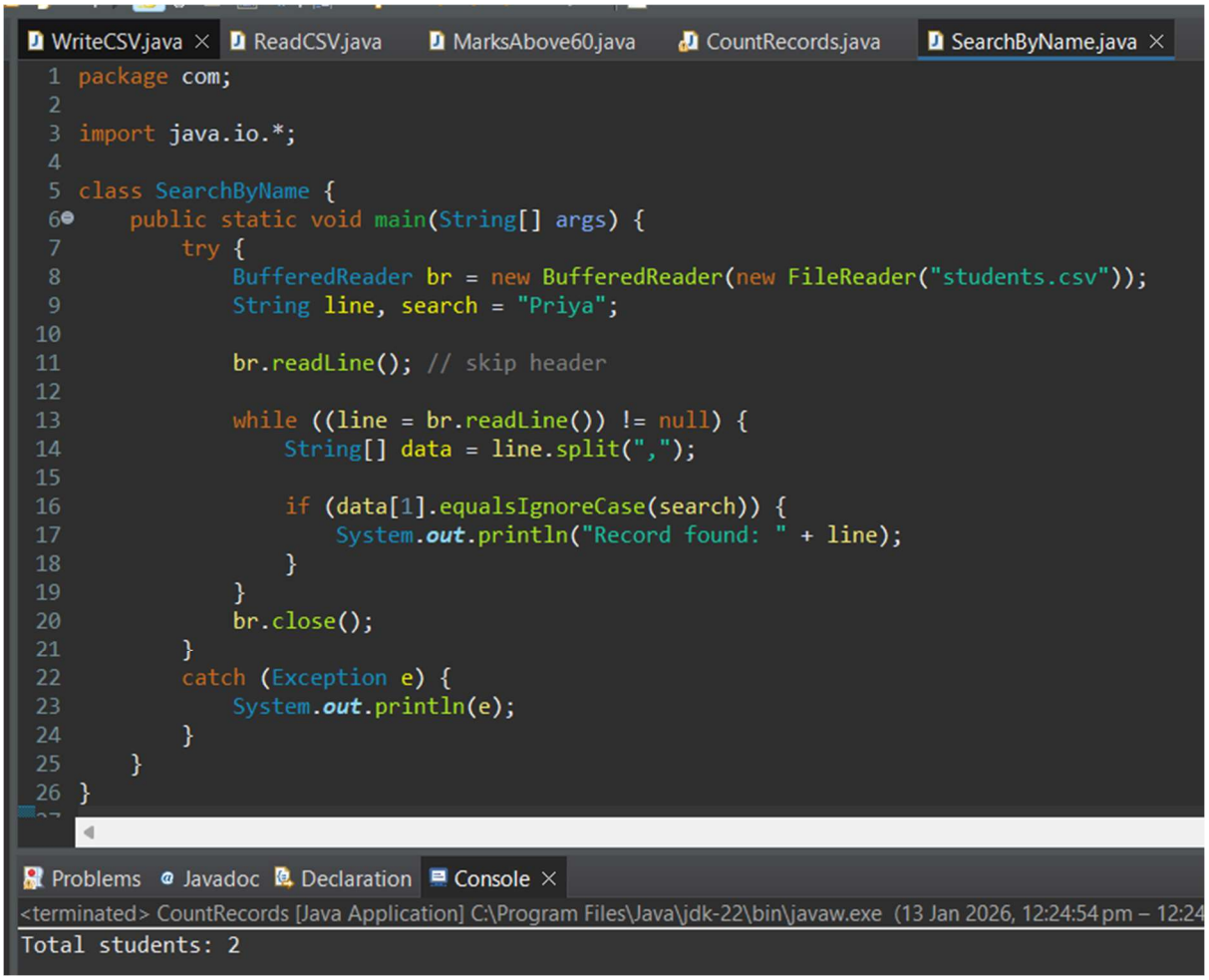
Conditions:

• Use equalsIgnoreCase()

• Display full record if name matches

Expected Outcome:

Matching student details printed.



Assignment 6

Title: Display Students Who Failed

Problem Statement:

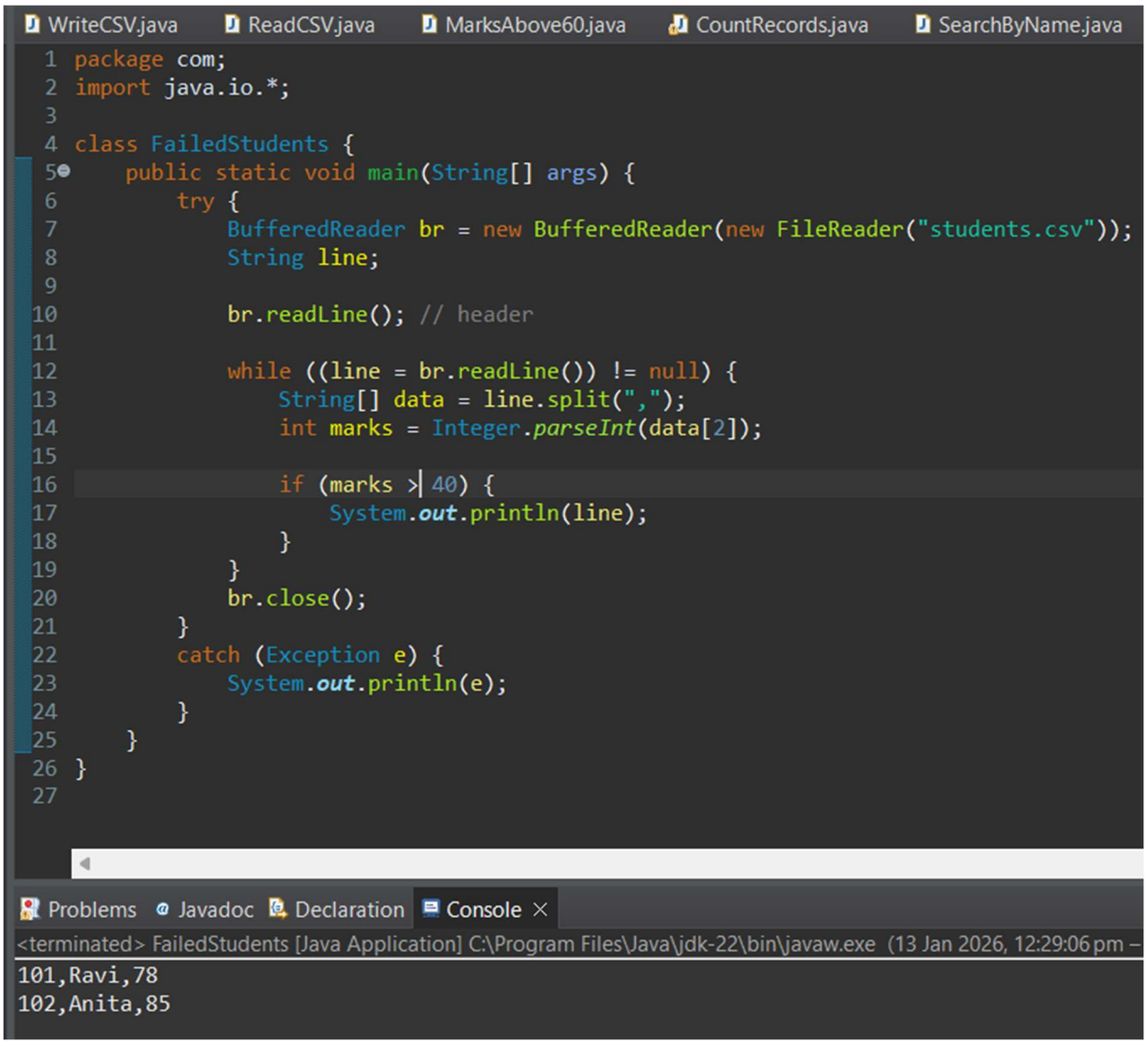
Display students whose marks are less than 40.

Conditions:

• Use if (marks < 40)• Read from CSV file

Expected Outcome:

Only failed students printed.



Assignment 7

Title: Calculate Average Marks

Problem Statement:

Read marks from students.csv and calculate average marks.

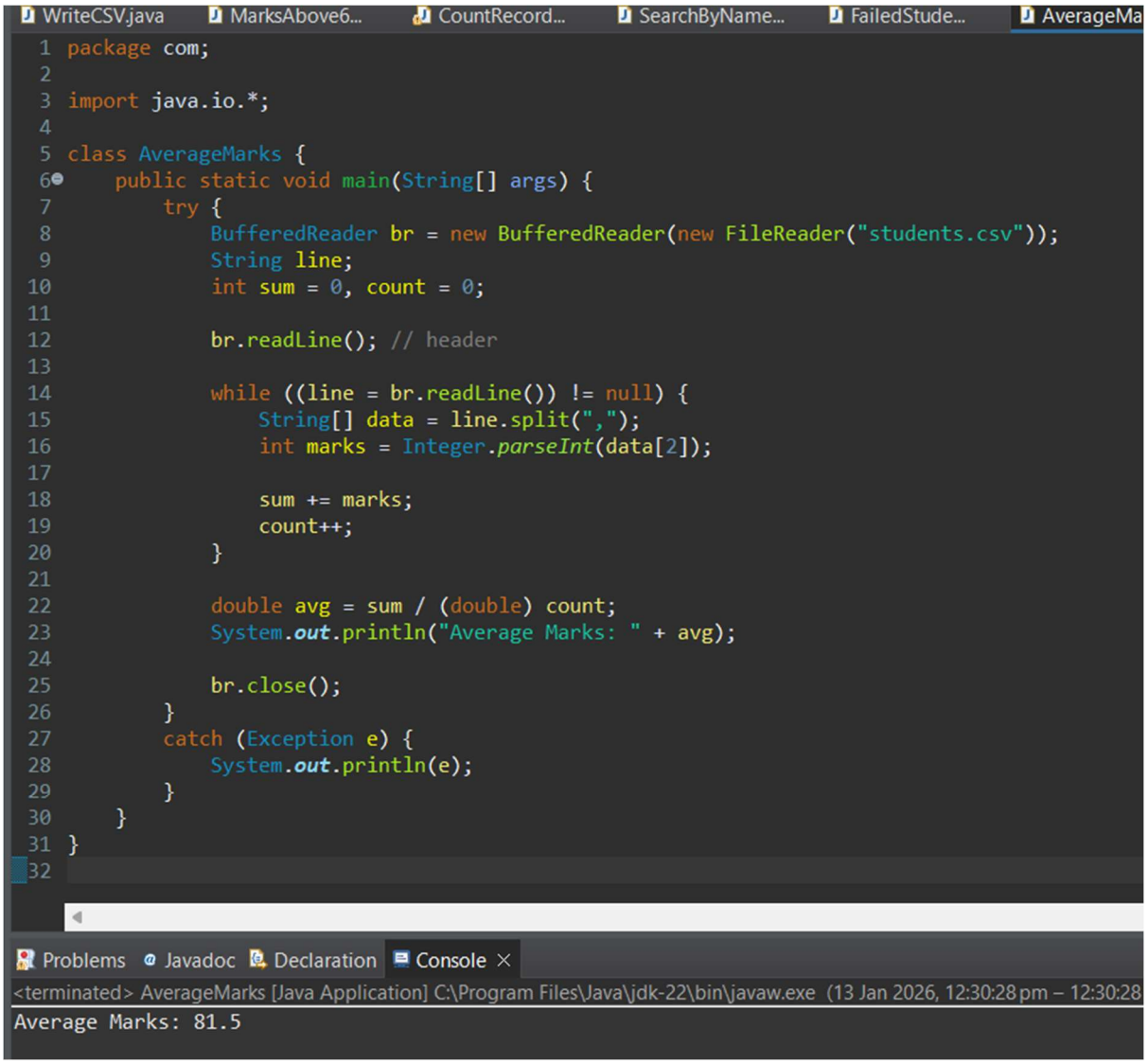
Conditions:

• Sum all marks

• Divide by number of students

Expected Outcome:

Average marks displayed.



Assignment 8

Title: Copy Passed Students to New CSV File

Problem Statement:

Read students.csv and write students with marks ≥ 50 into passed_students.csv.

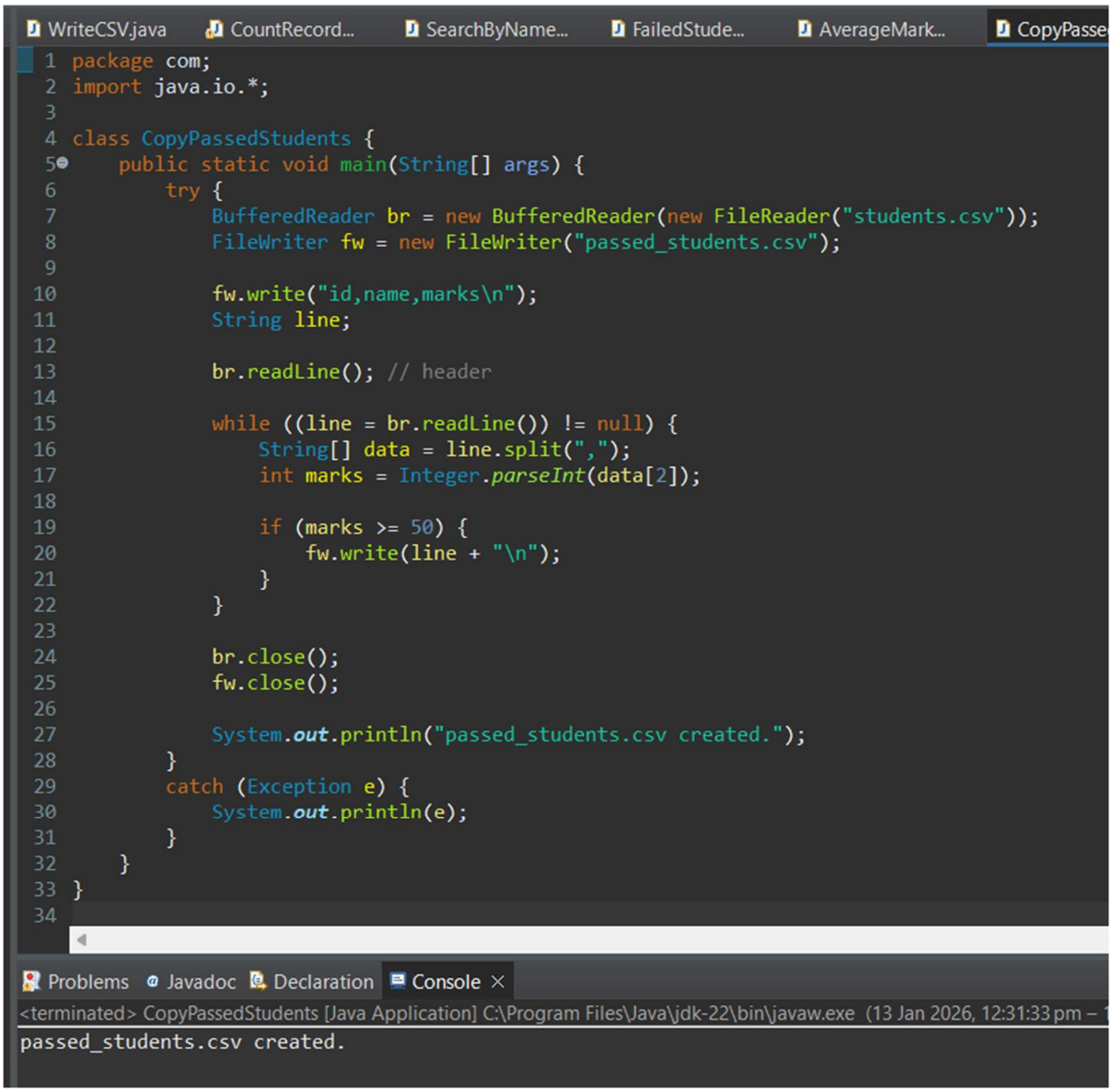
Conditions:

• Use FileWriter

• Maintain CSV format

Expected Outcome:

New CSV file created with passed students only.



Assignment 9

Title: Validate CSV Data While Reading

Problem Statement:

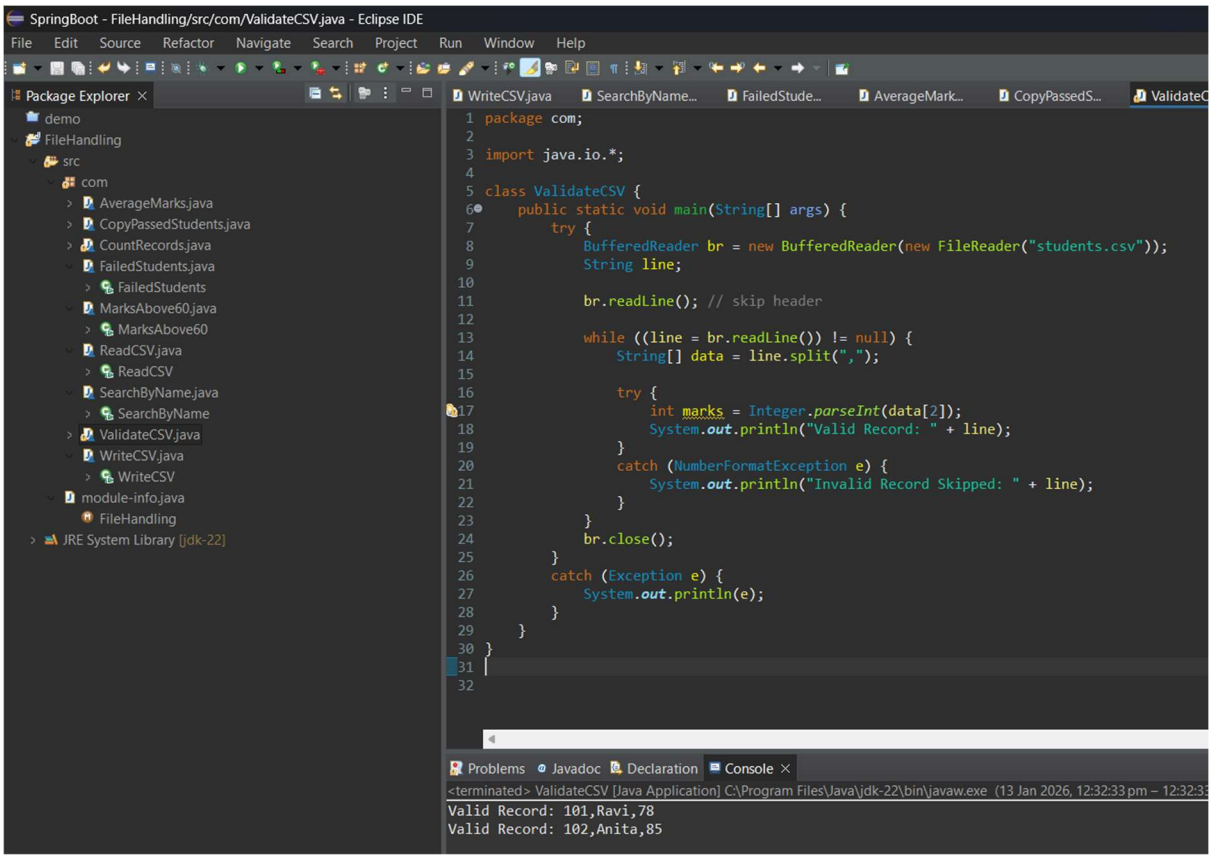
Skip records where marks field is empty or invalid.

Conditions:

• Use try-catch for NumberFormatException• Ignore invalid rows

Expected Outcome:

Only valid records processed.



Assignment 10

Title: Display Topper Details

Problem Statement:

Find and display the student with highest marks from students.csv.

Conditions:

• Compare marks

• Track maximum value

Expected Outcome:

Topper name and marks printed.

Sample CSV File (students.csv)

id,name,marks

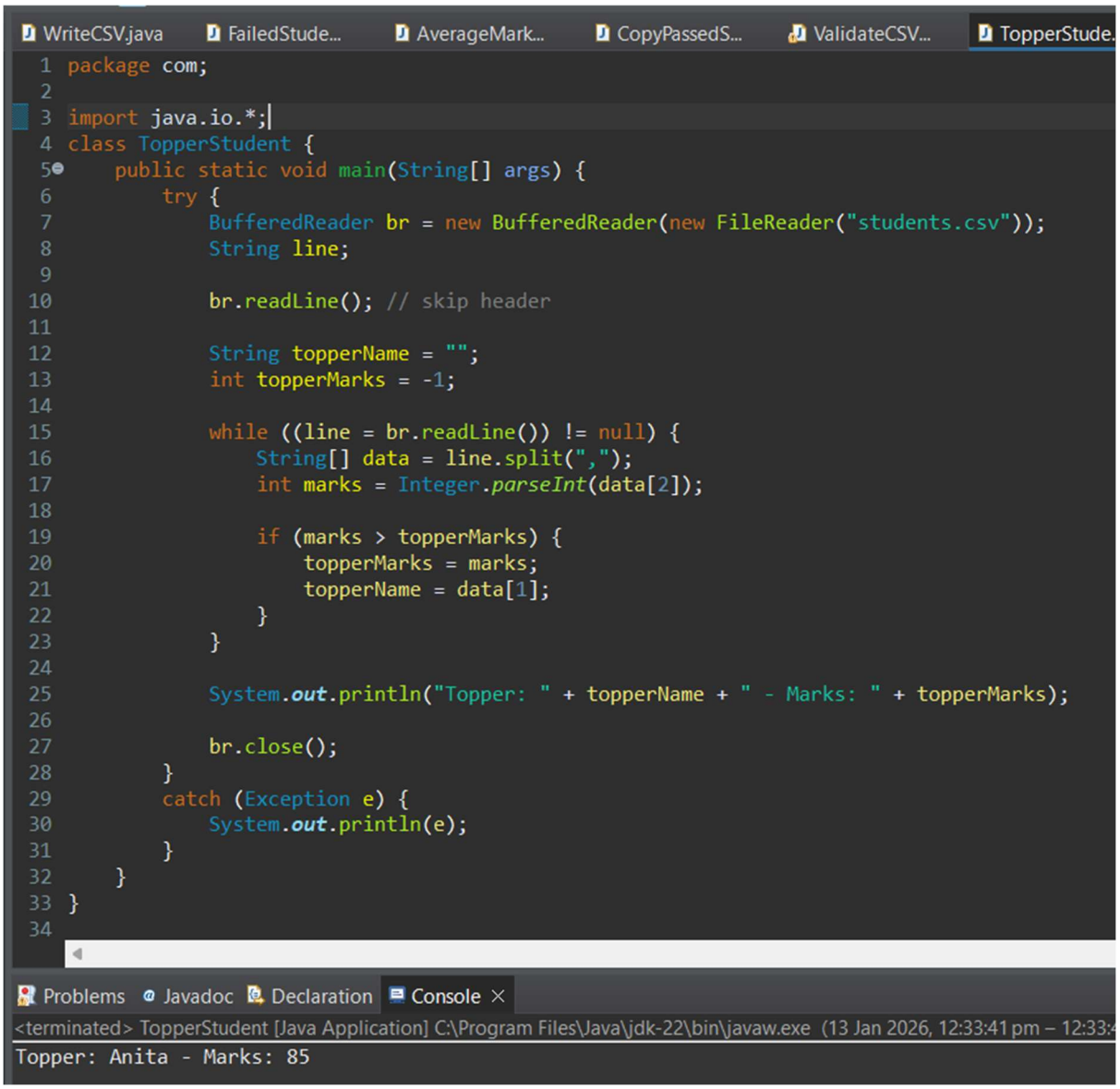
101,Ravi,78

102,Anita,85

103,Sunil,35

104,Priya,92

105,Aman,45


WriteCSV.java
FailedStude...
AverageMark...
CopyPassedS...
ValidateCSV...
TopperStude...
1 package com;
2
3 import java.io.*;
4 class TopperStudent {
5 public static void main(String[] args) {
6 try {
7 BufferedReader br = new BufferedReader(new FileReader("students.csv"));
8 String line;
9
10 br.readLine(); // skip header
11
12 String topperName = "";
13 int topperMarks = -1;
14
15 while ((line = br.readLine()) != null) {
16 String[] data = line.split(",");
17 int marks = Integer.parseInt(data[2]);
18
19 if (marks > topperMarks) {
20 topperMarks = marks;
21 topperName = data[1];
22 }
23 }
24
25 System.out.println("Topper: " + topperName + " - Marks: " + topperMarks);
26
27 br.close();
28 }
29 catch (Exception e) {
30 System.out.println(e);
31 }
32 }
33 }
34
Problems
Javadoc
Declaration
Console
<terminated> TopperStudent [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe (13 Jan 2026, 12:33:41 pm – 12:33:
Topper: Anita - Marks: 85