

ASSIGNMENT

B. LIKHITHA

Assignment 1

Title: Write Data to CSV File

Problem Statement:

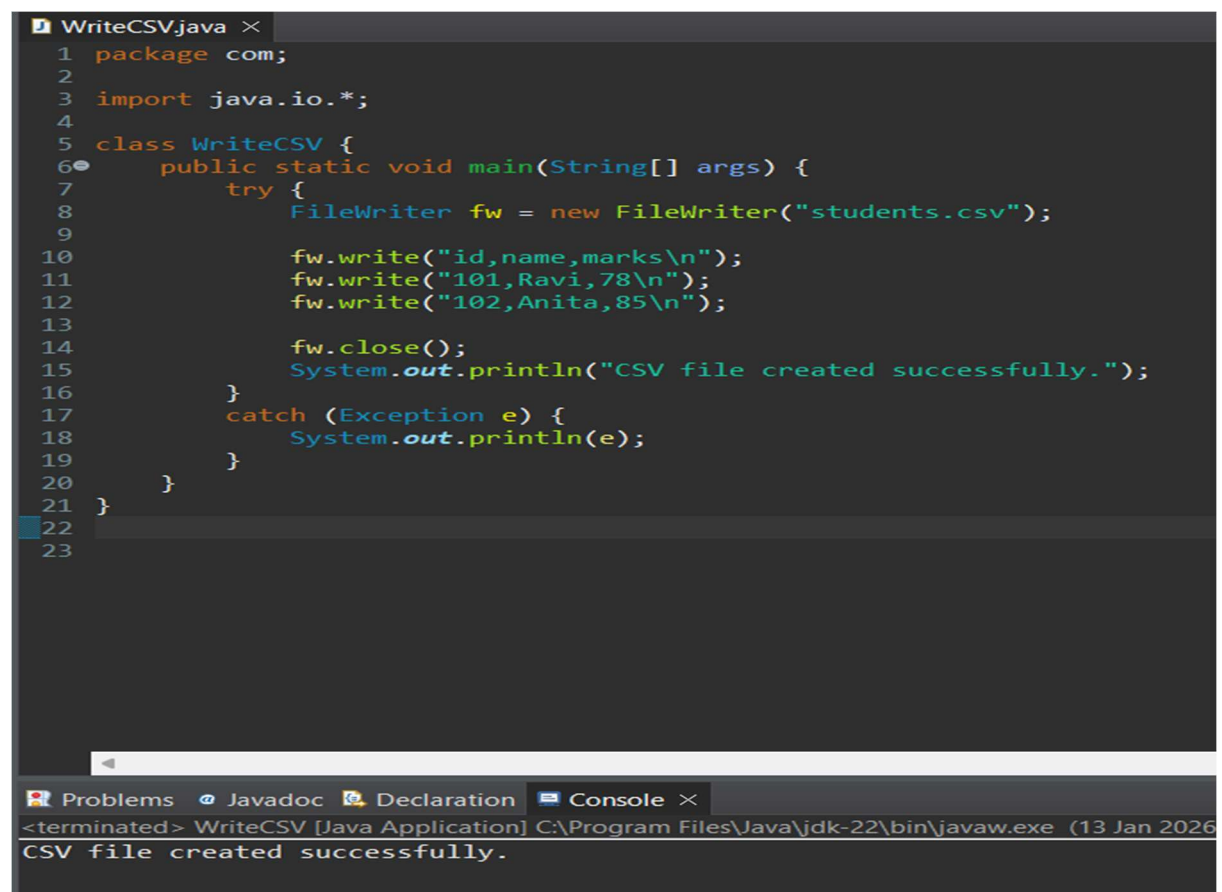
Write student details (id, name, marks) into a CSV file named students.csv.

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.



```
1 package com;
2
3 import java.io.*;
4
5 class WriteCSV {
6     public static void main(String[] args) {
7         try {
8             FileWriter fw = new FileWriter("students.csv");
9
10            fw.write("id,name,marks\n");
11            fw.write("101,Ravi,78\n");
12            fw.write("102,Anita,85\n");
13
14            fw.close();
15            System.out.println("CSV file created successfully.");
16        }
17        catch (Exception e) {
18            System.out.println(e);
19        }
20    }
21 }
22
23
```

Problems Javadoc Declaration Console

<terminated> WriteCSV [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe (13 Jan 2026)

CSV file created successfully.

Assignment 2

Title: Read CSV File and Display Data

Problem Statement:

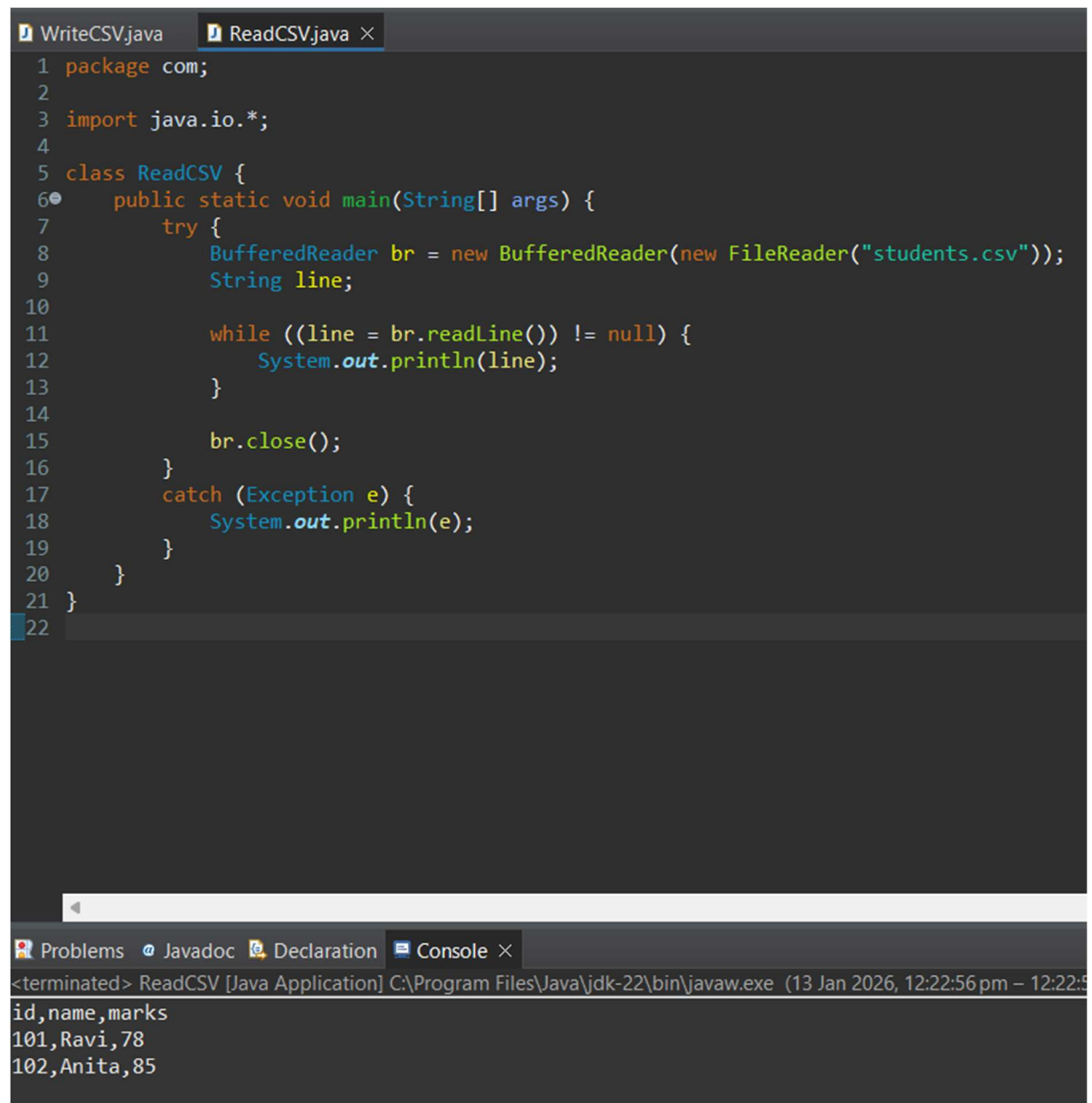
Read all records from students.csv and display them on the console.

Conditions:

- Use FileReader and BufferedReader
- Split each line using comma (,)

Expected Outcome:

All rows should be printed.



The screenshot shows an IDE with two tabs: 'WriteCSV.java' and 'ReadCSV.java'. The 'ReadCSV.java' tab is active, displaying the following Java code:

```
1 package com;
2
3 import java.io.*;
4
5 class ReadCSV {
6     public static void main(String[] args) {
7         try {
8             BufferedReader br = new BufferedReader(new FileReader("students.csv"));
9             String line;
10
11             while ((line = br.readLine()) != null) {
12                 System.out.println(line);
13             }
14
15             br.close();
16         }
17         catch (Exception e) {
18             System.out.println(e);
19         }
20     }
21 }
22
```

Below the code editor, the 'Console' tab is active, showing the output of the program:

```
<terminated> ReadCSV [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe (13 Jan 2026, 12:22:56 pm - 12:22:56 pm)
id,name,marks
101,Ravi,78
102,Anita,85
```

Assignment 3

Title: Display Students with Marks Greater Than 60

Problem Statement:

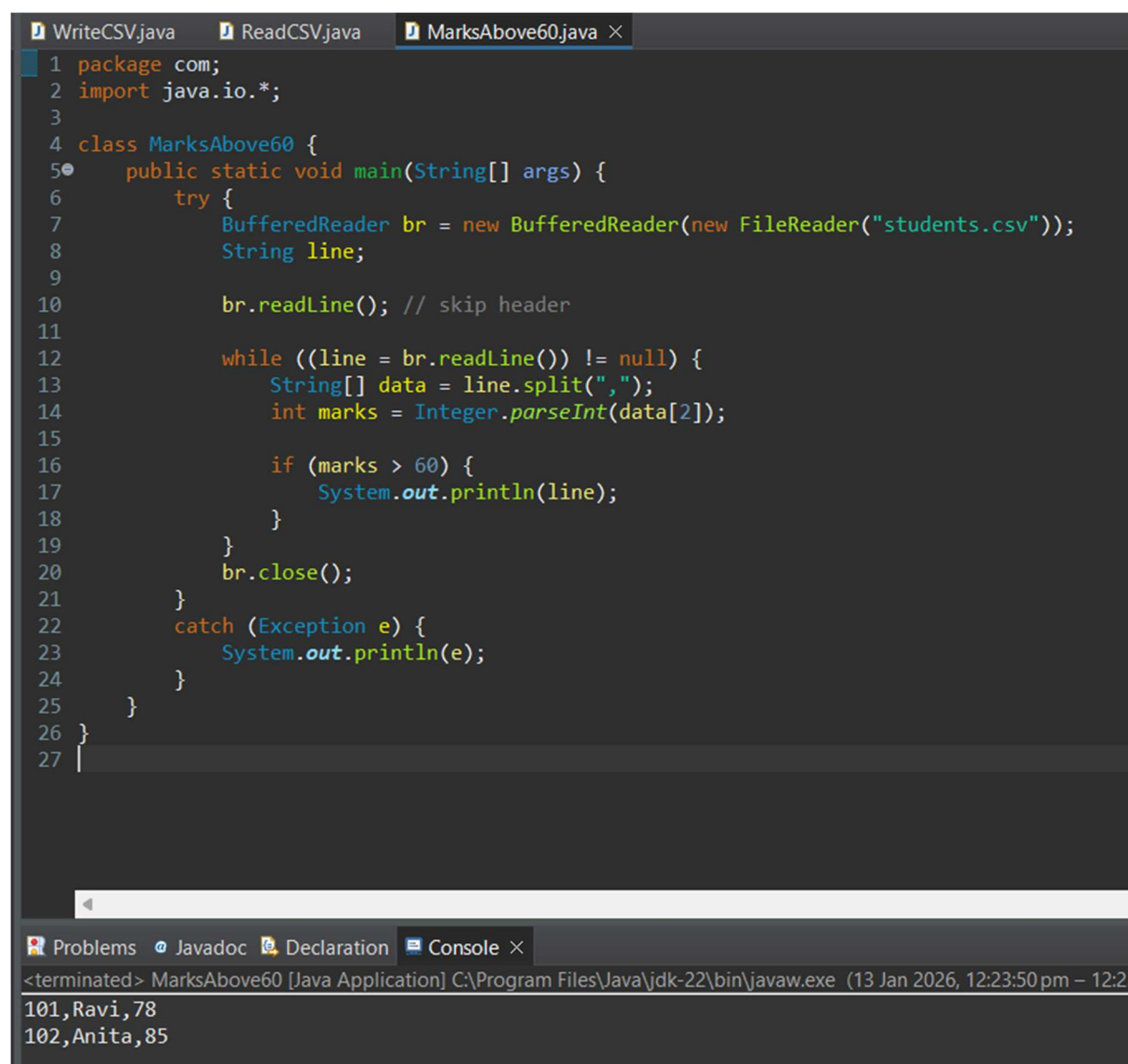
Read students.csv and display only students whose marks are greater than 60.

Conditions:

- Convert marks to integer
- Use if condition

Expected Outcome:

Only eligible students should be printed.



```
1 package com;
2 import java.io.*;
3
4 class MarksAbove60 {
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            while ((line = br.readLine()) != null) {
13                String[] data = line.split(",");
14                int marks = Integer.parseInt(data[2]);
15
16                if (marks > 60) {
17                    System.out.println(line);
18                }
19            }
20            br.close();
21        }
22        catch (Exception e) {
23            System.out.println(e);
24        }
25    }
26 }
27
```

Problems Javadoc Declaration Console ×

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

Assignment 4

Title: Count Number of Records in CSV File

Problem Statement:

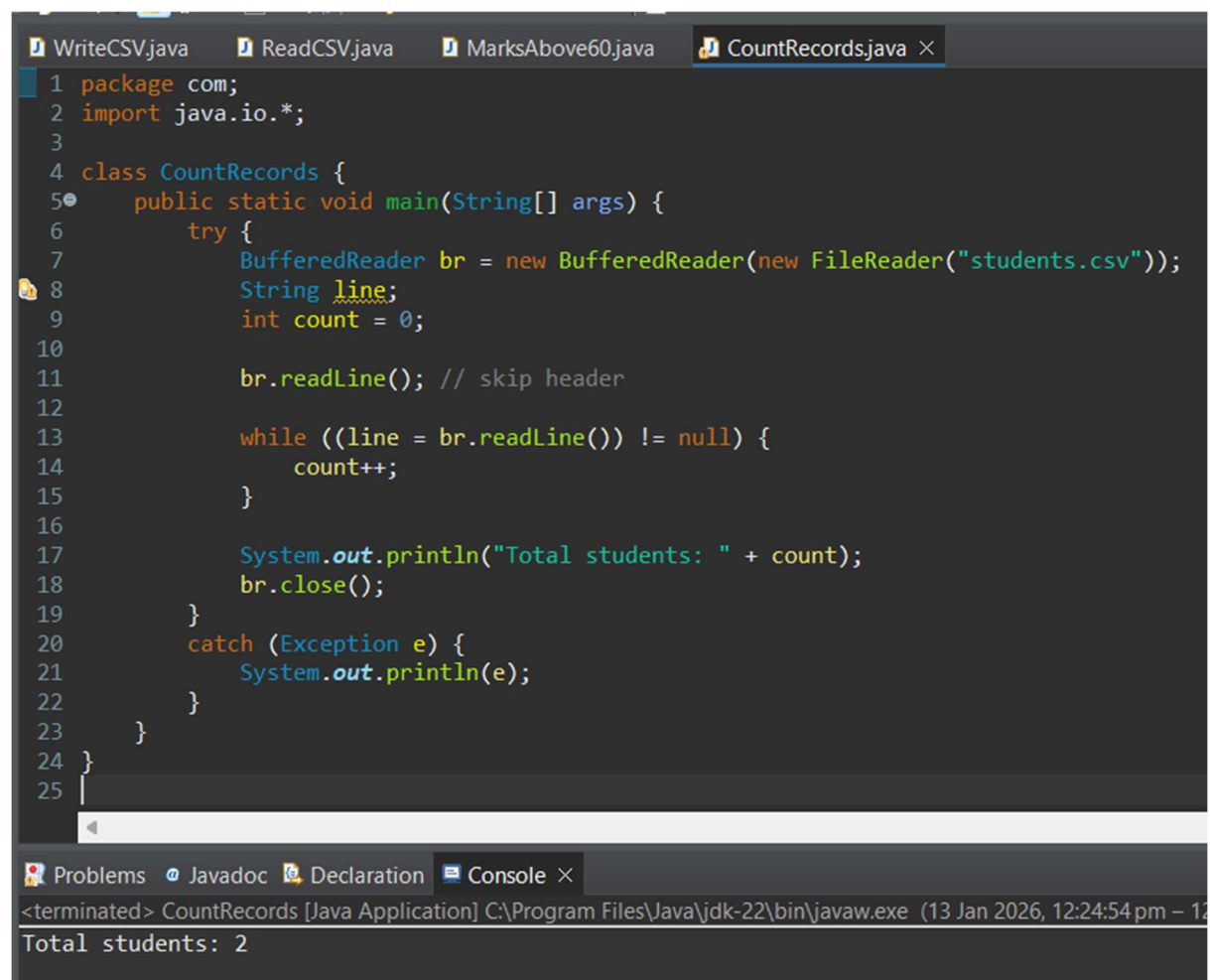
Count how many student records are present in students.csv.

Conditions:

- Ignore header line
- Use counter variable

Expected Outcome:

Total number of students displayed.



```
1 package com;
2 import java.io.*;
3
4 class CountRecords {
5     public static void main(String[] args) {
6         try {
7             BufferedReader br = new BufferedReader(new FileReader("students.csv"));
8             String line;
9             int count = 0;
10
11             br.readLine(); // skip header
12
13             while ((line = br.readLine()) != null) {
14                 count++;
15             }
16
17             System.out.println("Total students: " + count);
18             br.close();
19         }
20         catch (Exception e) {
21             System.out.println(e);
22         }
23     }
24 }
25
```

Problems Javadoc Declaration Console X

<terminated> CountRecords [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe (13 Jan 2026, 12:24:54 pm - 12)

Total students: 2

Assignment 5

Title: Search Student by Name

Problem Statement:

Search a student by name from students.csv.

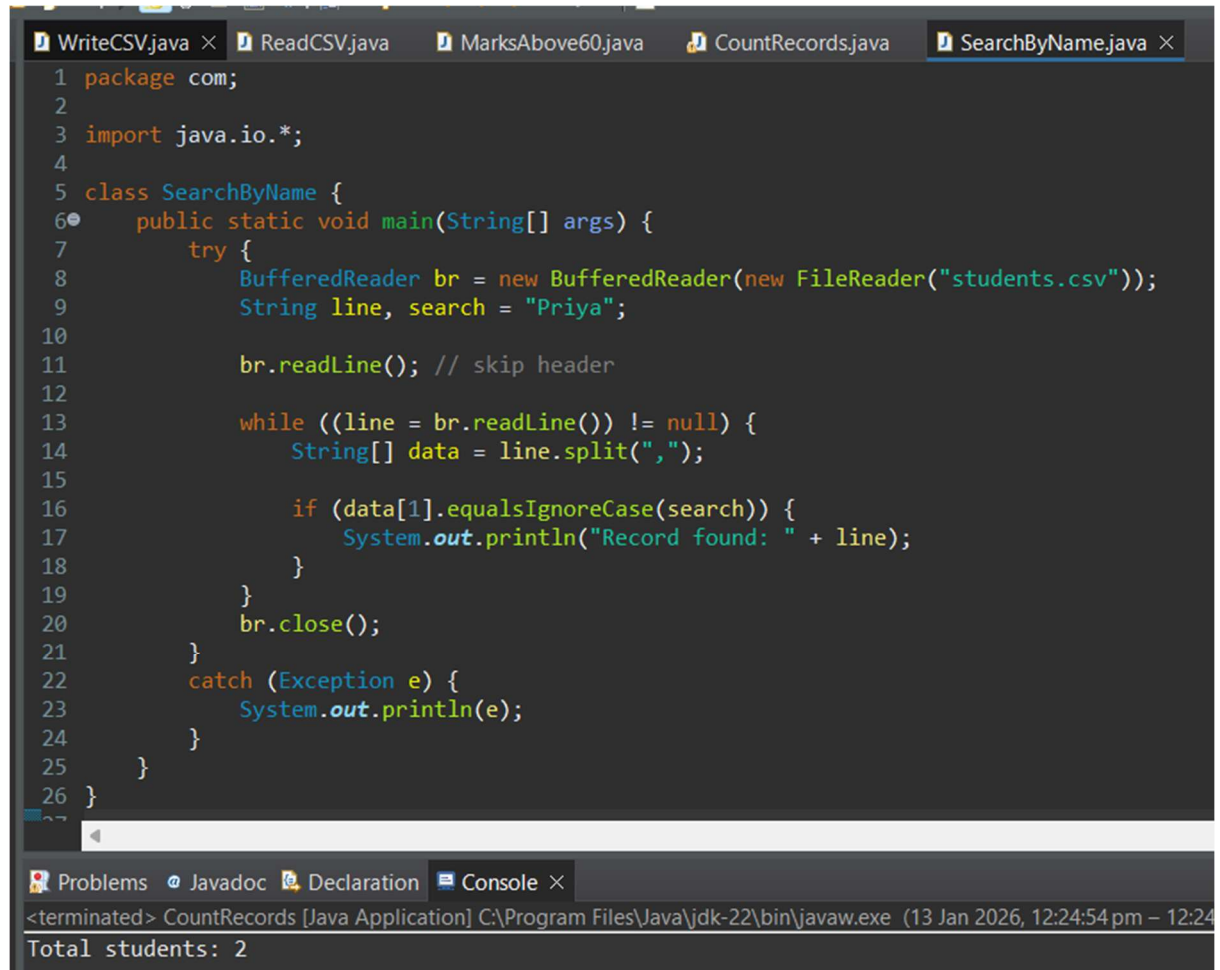
Conditions:

- Use equalsIgnoreCase()

- Display full record if name matches

Expected Outcome:

Matching student details printed.



```
1 package com;
2
3 import java.io.*;
4
5 class SearchByName {
6     public static void main(String[] args) {
7         try {
8             BufferedReader br = new BufferedReader(new FileReader("students.csv"));
9             String line, search = "Priya";
10
11             br.readLine(); // skip header
12
13             while ((line = br.readLine()) != null) {
14                 String[] data = line.split(",");
15
16                 if (data[1].equalsIgnoreCase(search)) {
17                     System.out.println("Record found: " + line);
18                 }
19             }
20             br.close();
21         }
22         catch (Exception e) {
23             System.out.println(e);
24         }
25     }
26 }
```

Problems Javadoc Declaration Console ×

<terminated> CountRecords [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe (13 Jan 2026, 12:24:54 pm – 12:24:54 pm)
Total students: 2

Assignment 6

Title: Display Students Who Failed

Problem Statement:

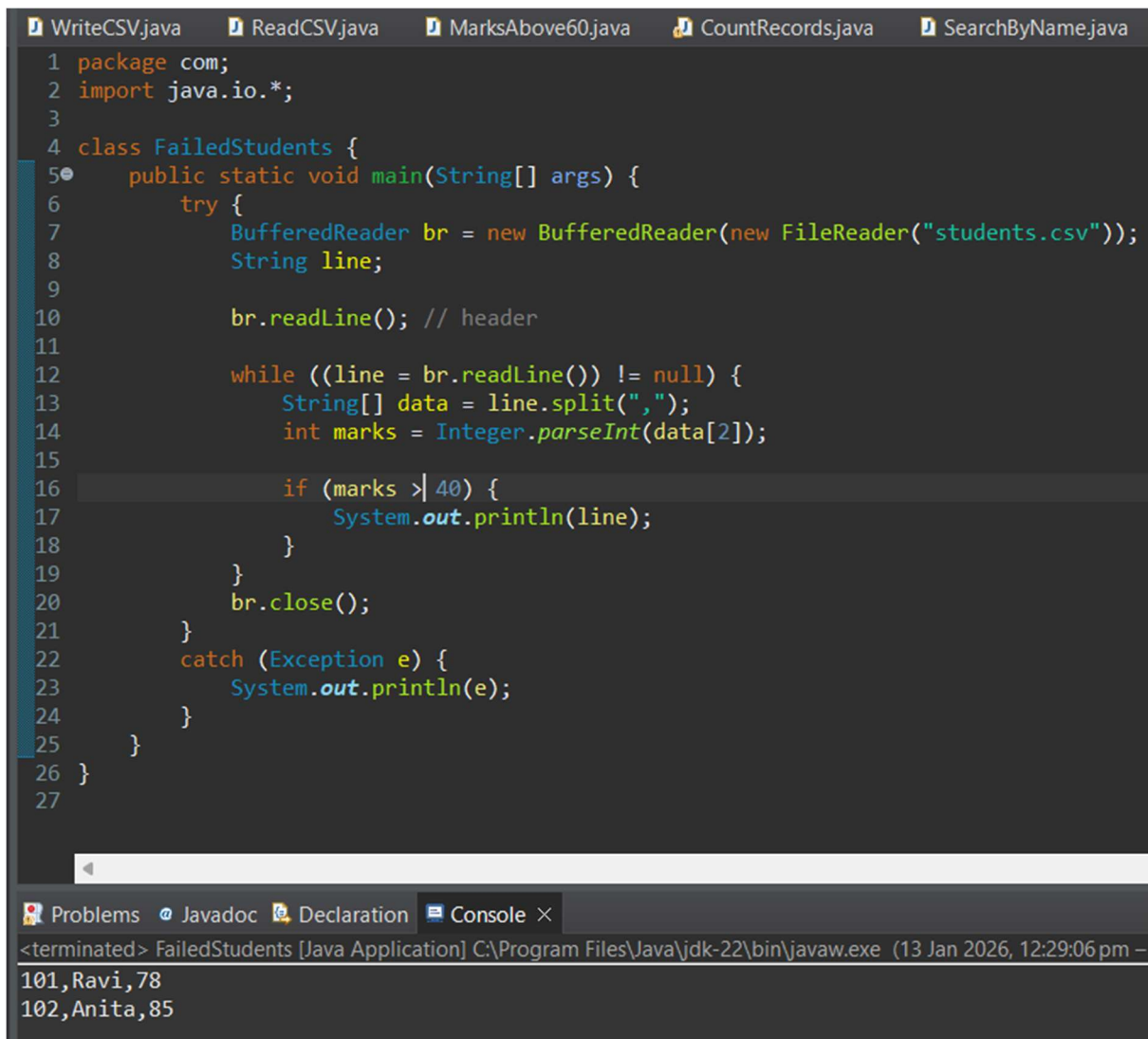
Display students whose marks are less than 40.

Conditions:

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

Expected Outcome:

Only failed students printed.



The screenshot shows an IDE with a Java file named `FailedStudents.java` and a console window. The code reads a CSV file `students.csv` and prints lines where the mark is greater than 40. The console output shows two lines: `101,Ravi,78` and `102,Anita,85`.

```
1 package com;
2 import java.io.*;
3
4 class FailedStudents {
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(); // header
11
12            while ((line = br.readLine()) != null) {
13                String[] data = line.split(",");
14                int marks = Integer.parseInt(data[2]);
15
16                if (marks > 40) {
17                    System.out.println(line);
18                }
19            }
20            br.close();
21        }
22        catch (Exception e) {
23            System.out.println(e);
24        }
25    }
26 }
27
```

Problems Javadoc Declaration Console ×

<terminated> FailedStudents [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe (13 Jan 2026, 12:29:06 pm -

101,Ravi,78
102,Anita,85

Assignment 7

Title: Calculate Average Marks

Problem Statement:

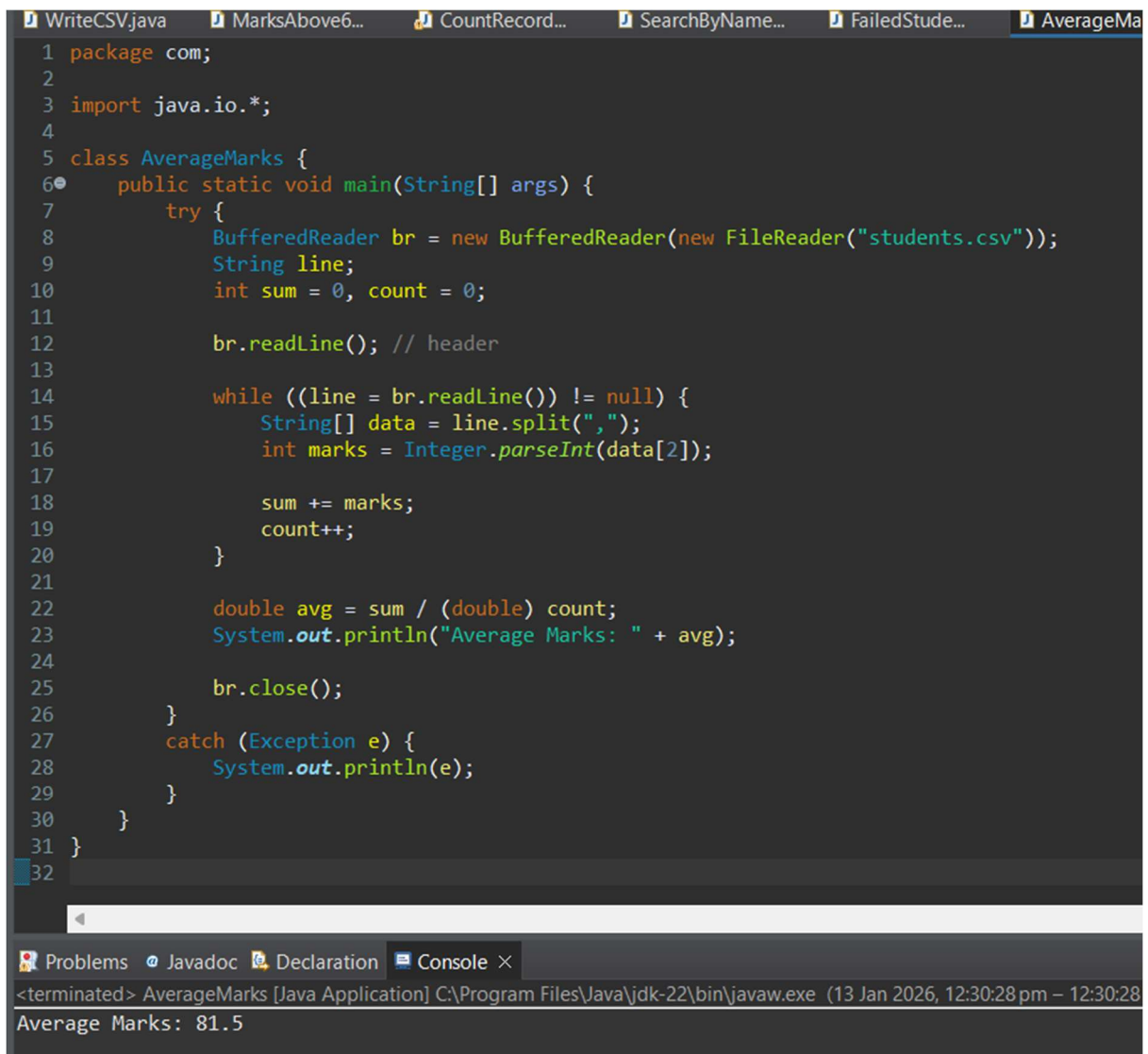
Read marks from students.csv and calculate average marks.

Conditions:

- Sum all marks
- Divide by number of students

Expected Outcome:

Average marks displayed.



```
1 package com;
2
3 import java.io.*;
4
5 class AverageMarks {
6     public static void main(String[] args) {
7         try {
8             BufferedReader br = new BufferedReader(new FileReader("students.csv"));
9             String line;
10            int sum = 0, count = 0;
11
12            br.readLine(); // header
13
14            while ((line = br.readLine()) != null) {
15                String[] data = line.split(",");
16                int marks = Integer.parseInt(data[2]);
17
18                sum += marks;
19                count++;
20            }
21
22            double avg = sum / (double) count;
23            System.out.println("Average Marks: " + avg);
24
25            br.close();
26        }
27        catch (Exception e) {
28            System.out.println(e);
29        }
30    }
31 }
32
```

Problems Javadoc Declaration Console ×

<terminated> AverageMarks [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe (13 Jan 2026, 12:30:28 pm – 12:30:28)

Average Marks: 81.5

Assignment 8

Title: Copy Passed Students to New CSV File

Problem Statement:

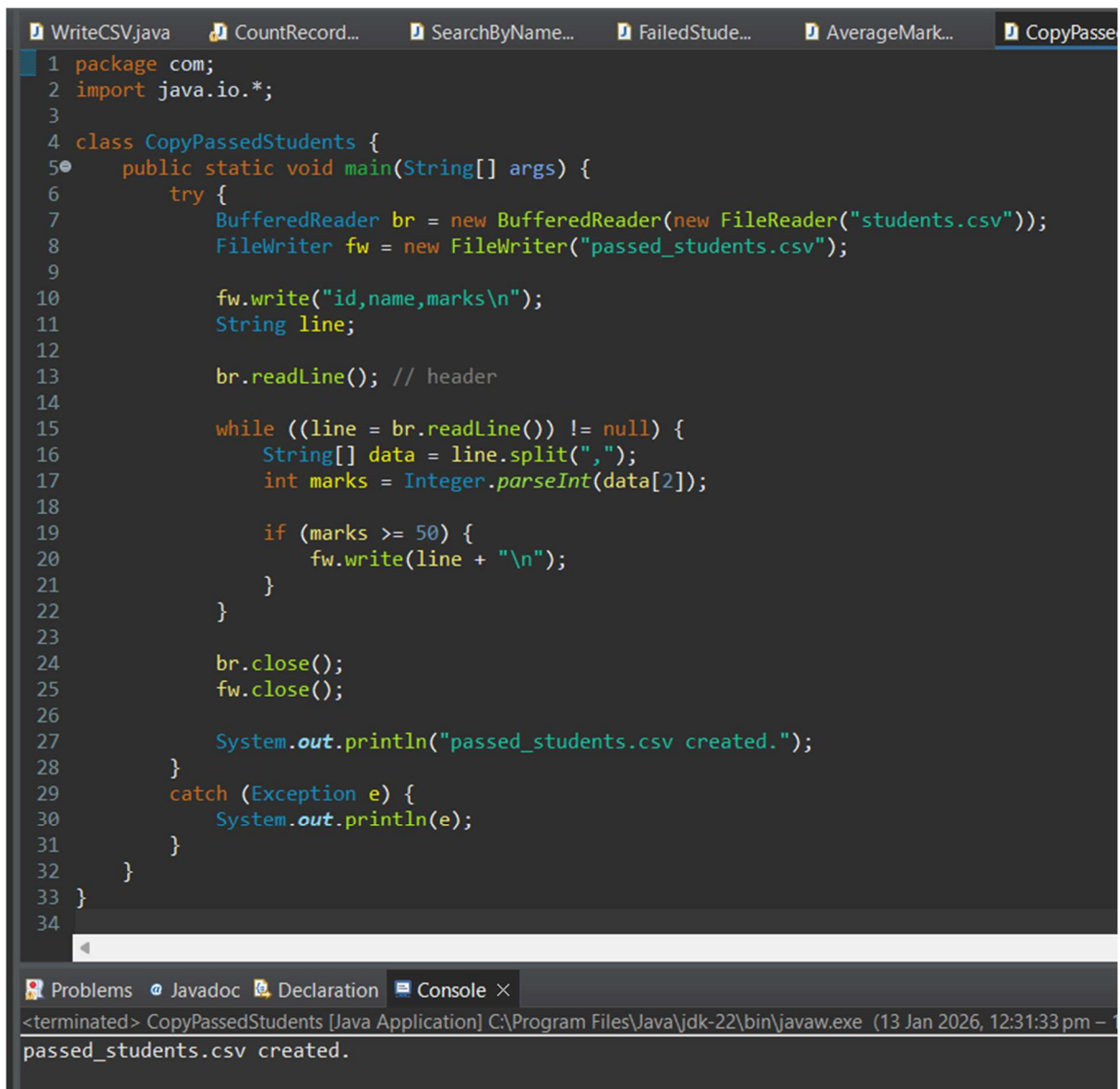
Read students.csv and write students with marks ≥ 50 into passed_students.csv.

Conditions:

- Use FileWriter
- Maintain CSV format

Expected Outcome:

New CSV file created with passed students only.



```
1 package com;
2 import java.io.*;
3
4 class CopyPassedStudents {
5     public static void main(String[] args) {
6         try {
7             BufferedReader br = new BufferedReader(new FileReader("students.csv"));
8             FileWriter fw = new FileWriter("passed_students.csv");
9
10            fw.write("id,name,marks\n");
11            String line;
12
13            br.readLine(); // header
14
15            while ((line = br.readLine()) != null) {
16                String[] data = line.split(",");
17                int marks = Integer.parseInt(data[2]);
18
19                if (marks >= 50) {
20                    fw.write(line + "\n");
21                }
22            }
23
24            br.close();
25            fw.close();
26
27            System.out.println("passed_students.csv created.");
28        }
29        catch (Exception e) {
30            System.out.println(e);
31        }
32    }
33 }
34
```

Problems Javadoc Declaration Console X

<terminated> CopyPassedStudents [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe (13 Jan 2026, 12:31:33 pm - 1
passed_students.csv created.

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.


```
1 package com;
2
3 import java.io.*;
4
5 class ValidateCSV {
6     public static void main(String[] args) {
7         try {
8             BufferedReader br = new BufferedReader(new FileReader("students.csv"));
9             String line;
10
11             br.readLine(); // skip header
12
13             while ((line = br.readLine()) != null) {
14                 String[] data = line.split(",");
15
16                 try {
17                     int marks = Integer.parseInt(data[2]);
18                     System.out.println("Valid Record: " + line);
19                 }
20                 catch (NumberFormatException e) {
21                     System.out.println("Invalid Record Skipped: " + line);
22                 }
23             }
24             br.close();
25         }
26         catch (Exception e) {
27             System.out.println(e);
28         }
29     }
30 }
31
32
```

Problems Javadoc Declaration Console

<terminated> ValidateCSV [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe (13 Jan 2026, 12:32:33 pm - 12:32:33 pm)

Valid Record: 101,Ravi,78

Valid Record: 102,Anita,85

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

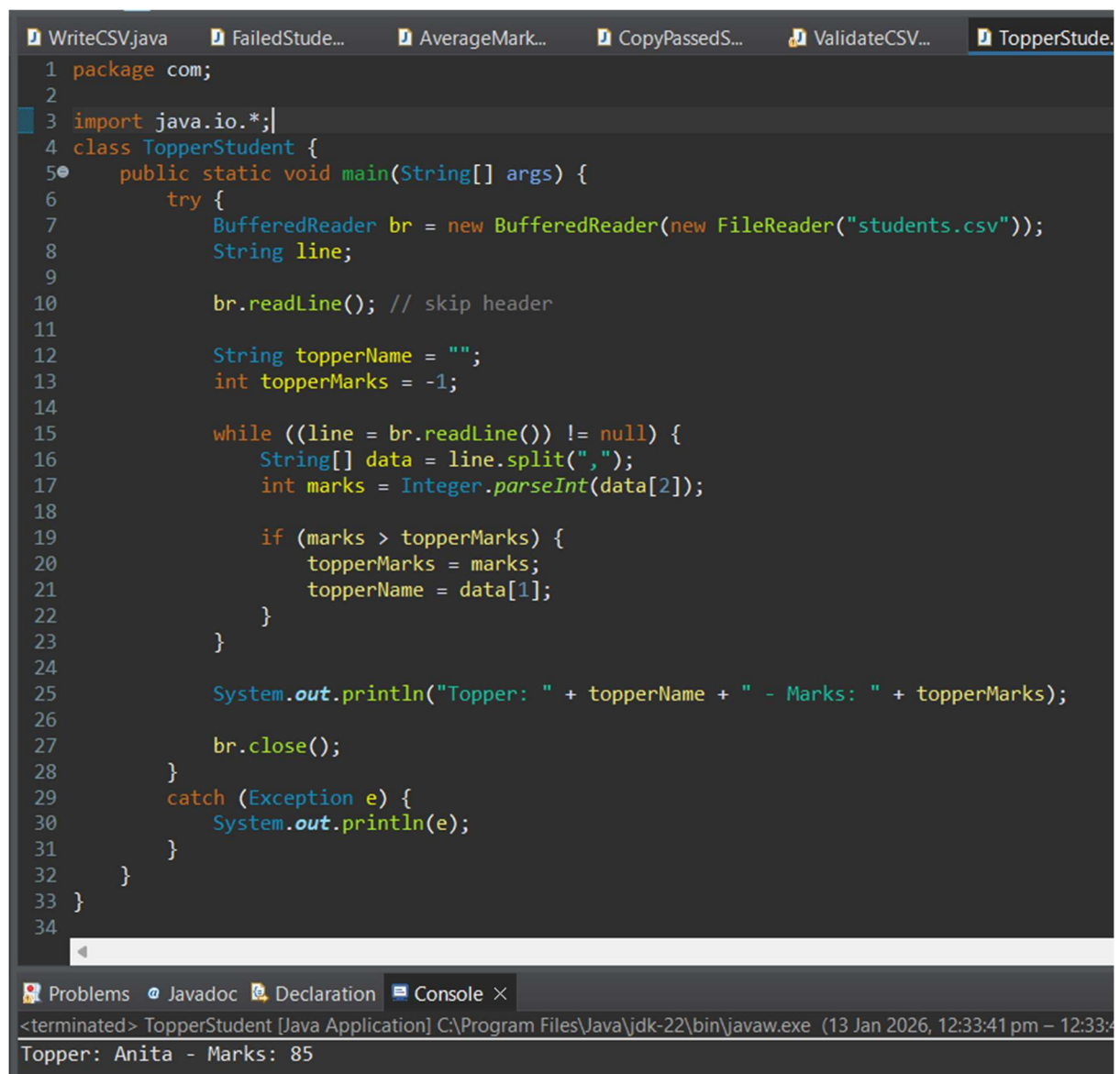
101,Ravi,78

102,Anita,85

103,Sunil,35

104,Priya,92

105,Aman,45



```
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 – 12:33:41 pm)
Topper: Anita - Marks: 85