Kelvin Ngeno

AIT 502 – M2B

Fall 2025

Program: Student Performance Analyzer

Program Requirements

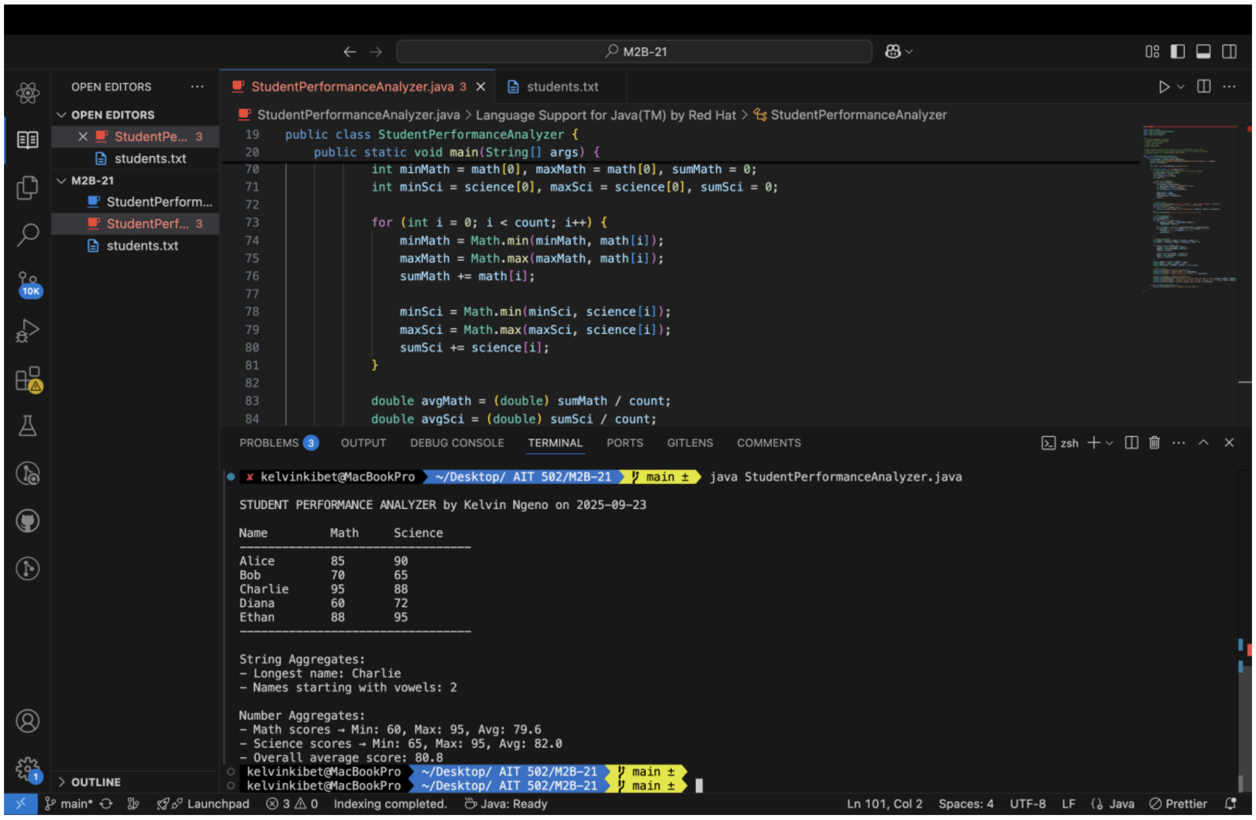
This program reads a list of students and their scores from a text file. Each line of the file contains a student name followed by two integers (for example, math and science scores). The file ends with a sentinel marker ###END###. The program stores the names in one array and the scores in two parallel arrays, so that the arrays are index-linked and have the same size. Once the data is read, the program prints a professional-looking table showing each student's name and scores. Finally, it performs several aggregate calculations: two on the string data (longest name and count of names starting with vowels) and three on the numeric data (min, max, and average for math and science scores, plus the overall average).

Why the Program is Relevant to the Topic Area

This program is relevant because it demonstrates the core skills emphasized in this section: reading structured data from a file, storing it in arrays, and applying both string-based and number-based aggregate operations. The assignment requires index-linked arrays and matrix-style numeric data, which are foundational concepts for working with tabular information in Java. The program also aligns with real-world applications such as analyzing grades, generating reports, and performing descriptive statistics. It is directly tied to the topic area of file handling, arrays, and data aggregation.

Detailed Explanation of How the Problem Was Solved

The program begins by printing a header line with the program name, student name, and current date. It then opens the file students.txt using a Scanner. Inside a loop, the program reads names and two scores until it encounters the sentinel ###END###. Names are stored in a String[], and the scores are stored in two parallel int[] arrays, ensuring index-linking. Once all input is read, the program prints a neatly formatted table of names and scores using printf. Next, it computes aggregates. For strings, it finds the longest name and counts how many names start with a vowel. For numbers, it calculates the minimum, maximum, and average for both math and science scores, and also the overall average across both subjects. Finally, the program prints the aggregate results with clear explanatory labels.



```java
public class StudentPerformanceAnalyzer {
    public static void main(String[] args) {
            int minMath = math[0], maxMath = math[0], sumMath = 0;
            int minSci = science[0], maxSci = science[0], sumSci = 0;

            for (int i = 0; i < count; i++) {
                minMath = Math.min(minMath, math[i]);
                maxMath = Math.max(maxMath, math[i]);
                sumMath += math[i];

                minSci = Math.min(minSci, science[i]);
                maxSci = Math.max(maxSci, science[i]);
                sumSci += science[i];
            }

            double avgMath = (double) sumMath / count;
            double avgSci = (double) sumSci / count;
```

```
kelvinkibet@MacBookPro ~/Desktop/ AIT 502/M2B-21 main ± java StudentPerformanceAnalyzer.java

STUDENT PERFORMANCE ANALYZER by Kelvin Ngeno on 2025-09-23

Name        Math     Science
---------------------------------
Alice       85       90
Bob         70       65
Charlie     95       88
Diana       60       72
Ethan       88       95
---------------------------------

String Aggregates:
- Longest name: Charlie
- Names starting with vowels: 2

Number Aggregates:
- Math scores → Min: 60, Max: 95, Avg: 79.6
- Science scores → Min: 65, Max: 95, Avg: 82.0
- Overall average score: 80.8
kelvinkibet@MacBookPro ~/Desktop/ AIT 502/M2B-21 main ±
kelvinkibet@MacBookPro ~/Desktop/ AIT 502/M2B-21 main ±
```