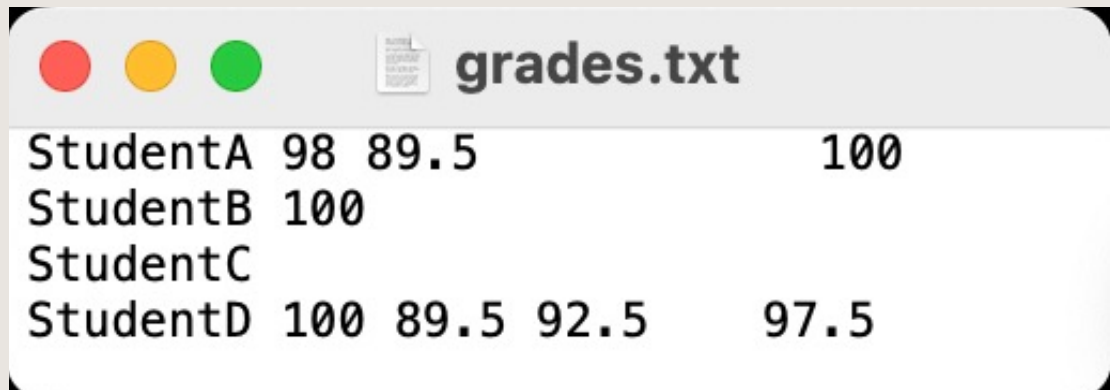


File Input (Line-Based Processing)

zyBook Chap 6.4

Coding Practice – Grades from File (Token-Based)

Write a program that adds up the grades from a file via token-based processing.

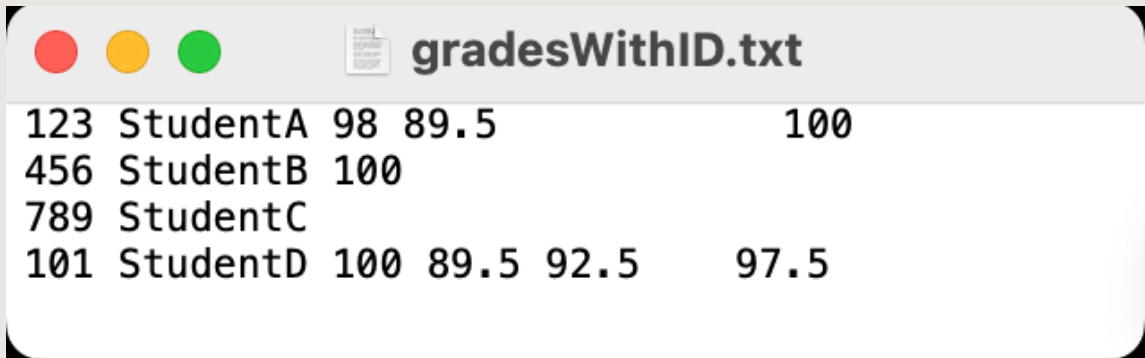


A screenshot of a text editor window titled "grades.txt". The window contains the following text:

StudentA	98	89.5		100
StudentB	100			
StudentC				
StudentD	100	89.5	92.5	97.5

```
$ javac GradeCalculator.java
$ java GradeCalculator
StudentA: 287.5
StudentB: 100.0
StudentC: 0.0
StudentD: 379.5
```

Brainstorm: Would the token-based processing work given the following input file?



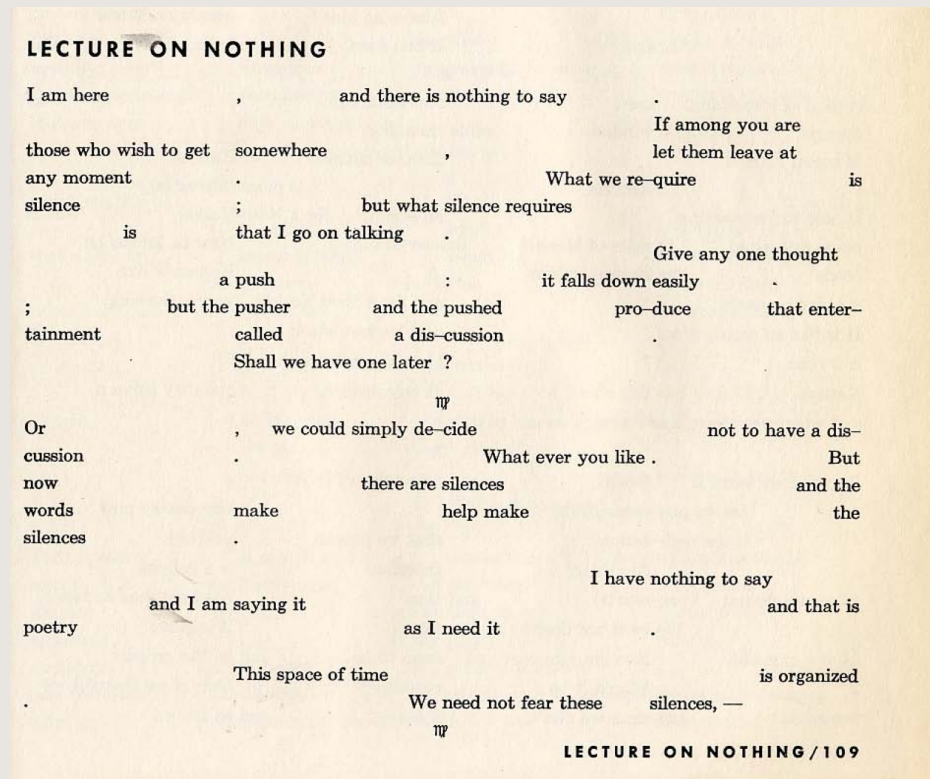
A screenshot of a text file named 'gradesWithID.txt'. The file contains four lines of data, each representing a student. The first line is '123 StudentA 98 89.5 100', the second is '456 StudentB 100', the third is '789 StudentC', and the fourth is '101 StudentD 100 89.5 92.5 97.5'. The file is displayed in a window with a title bar showing three colored circles (red, yellow, green) and the filename 'gradesWithID.txt'.

ID	Student	Grade 1	Grade 2	Grade 3	Grade 4
123	StudentA	98	89.5		100
456	StudentB	100			
789	StudentC				
101	StudentD	100	89.5	92.5	97.5

```
$ javac GradeCalculator.java
$ java GradeCalculator
123: 0.0
StudentA: 743.0
StudentB: 889.0
StudentC: 101.0
StudentD: 381.0
```

Line-Based Processing

- **Keeps white space and line breaks** of text being processed
 - Important for poems or formatted text



“Functional White”

Line-Based Processing

- **Step 1:** Construct a Scanner for the input file

```
Scanner fileScnr = new Scanner(new File(fileName));
```

- **Step 2:** While there's a line, read the entire line as a String

```
String line = fileScnr.nextLine();
```

- **Step 3:** Construct a Scanner to **tokenize the String** (each line)

```
Scanner lineScnr = new Scanner(line);
```

Line-Based Processing

```
// A scanner to process the file
Scanner fileScnr = new Scanner(new File(fileName));

// Check if there is one more line in the file
while (fileScnr.hasNextLine()) {
    // Scan in the line as a String
    String line = fileScnr.nextLine();

    // Tokenize a String with Scanner
    Scanner lineScnr = new Scanner(line);

    // Check if there is one more token in the line
    while (lineScnr.hasNext()) {
        // Consume the token
        lineScnr.next();
    }
    lineScnr.close(); // Close the scanner for the line
}
fileScnr.close(); // Close the scanner for the file
```

```

import java.io.*;
import java.util.Scanner;

public class GradeIDCalculator {
    public static void main (String[] args) throws FileNotFoundException {
        Scanner fileScnr = new Scanner(new File("gradesWithID.txt"));
        processGrade(fileScnr);
    }

    public static void processGrade(Scanner fileScnr) {
        // While there is one more line in the file
        while (fileScnr.hasNextLine()) {

            // Read the entire line as a String, where each line contains the info for one student
            String line = fileScnr.nextLine();

            // Tokenize the String by passing it as a parameter to a Scanner
            Scanner lineScnr = new Scanner(line);

            // The first token in each line is an int (student id)
            int stuID = lineScnr.nextInt();
            // The second token in each line is a String (student name)
            String stuName = lineScnr.next();

            double total = 0;
            // Add up all doubles in this line
            while (lineScnr.hasNextDouble()) {
                total += lineScnr.nextDouble();
            }
            lineScnr.close(); // Close the Scanner for the line

            System.out.println(stuName + " (id: " + stuID + "): " + total);
        }
        fileScnr.close(); // Close the Scanner for the input file
    }
}

```

Sample Solution

```

$ javac GradeIDCalculator.java
$ java GradeIDCalculator
StudentA (id: 123): 287.5
StudentB (id: 456): 100.0
StudentC (id: 789): 0.0
StudentD (id: 101): 379.5

```