**Narender Reddy Gopu**

**999901076**

**MCIS 5103: Advanced Programming Concepts**

**Architecture Diagram:**

Convert the numeric grade into letter grade

Displaying Grade and its count

GUI to display the final letter grade for each student by adding a scroll bar to fit all the names and grades.

Calculate the numeric grade

Take the input as text file containing student information and Split into tokens based on comma delimiter

**Requirement 1 : Ingesting student\_grades\_input.txt**

**Step 1:**

Using FileReader for reading the data from student\_grades\_input.txt.

bufferedReader = new BufferedReader(new FileReader("student\_grades\_input.txt"))

**Step 2:**

split() is used to split given input line based on comma delimiter into tokens.

line.split(",");

**Requirement 2 : Calculating the final numerical grade for each student**

**Step 1:**

Take the tokens array

**Step 2:**

Calculate homework value from 1st 3 tokens average and then multiplying by 0.45

double homework = 0.45 \* (((Integer.parseInt(tokens[1]) + Integer.parseInt(tokens[2]) + Integer.parseInt(tokens[3])))/3);

**Step 3:**

Calculate project from 5th token and multiply by 0.25

double project = 0.25 \* Integer.parseInt(tokens[5]);

**Step 4:**

Calculate midFinal from 4th and 6th tokens average and then multiplying by 0.30.

double midFinal = 0.30 \* (((Integer.parseInt(tokens[4]) + Integer.parseInt(tokens[6])))/2)

**Step 5:**

Calculate final numeric grade from step 2 to 5

int final\_numeric\_grade = (int) (homework + project + midFinal);

**Requirement 3 : Determining the final letter grade**

**Step 1:**

take the final numeric grade calculated in requirement 2

**Step 2 :**

calculate the letter grade and grade counts from A-F in 0-4 indices

char letter\_grade = 'F';

if(final\_numeric\_grade >= 90)

{

letter\_grade = 'A';

gradeCount[0]++;

}

else if(final\_numeric\_grade >= 80)

{

letter\_grade = 'B';

gradeCount[1]++;

}

else if(final\_numeric\_grade >= 70)

{

letter\_grade = 'C';

gradeCount[2]++;

}

else if(final\_numeric\_grade >= 60)

{

letter\_grade = 'D';

gradeCount[3]++;

}

else

{

gradeCount[4]++;

}

**Requirement 4 : GUI for display grades and counts**

**Step 1: GUI creation by creating frame and adding table and labels to it.**

JFrame frame = new JFrame("Final Grades Calculator");

frame.setLayout(new BorderLayout());

JTable table = new JTable(gradeData.toArray(new String[0][0]), new String[] { "Name of Student", "Grade of Student" });

**Step 2:**

Adding ScrollPane to frame

JScrollPane jsp = new JScrollPane(table);

frame.getContentPane().add(jsp);

**Requirement5:Displaying Grade and its count**

**Step1:**

Adding grade labels and its count at the bottom of frame

JLabel label = new JLabel("A:" + gradeCount[0] + " B:" + gradeCount[1] + " C:" + gradeCount[2] + " D:" + gradeCount[3] + " F:" + gradeCount[4], SwingConstants.CENTER);

frame.getContentPane().add(label, BorderLayout.PAGE\_END);

**Program:**

/\*

Name: Narender Reddy Gopu

ID: 999901076

Course Code: MCIS 5103

Course Name: Advanced Programming Concepts

\*/

import javax.swing.\*;

import java.awt.\*;

import java.io.\*;

import java.util.\*;

class FinalGradeCalculator

{

public static void main(String[] args)

{

BufferedReader bufferedReader = null;

ArrayList<String[]> gradeData = new ArrayList<>();

//grade counts array to handle 5 grades

int[] gradeCount = new int[] { 0, 0, 0, 0, 0 };

try

{

**//Requirement1:Ingesting student\_grades\_input.txt**

**//initialize buffered reader**

**bufferedReader = new BufferedReader(new FileReader("student\_grades\_input.txt"));**

//to skip the header

bufferedReader.readLine();

//Convert data to list of string array

String line = null;

//repeat until all lines are processed

while ((line = bufferedReader.readLine()) != null)

{

**//Requirement1:Comma delimited input line**

**String tokens[] = line.split(",");**

**//Requirement2:Computing the final numerical grade for each student**

**double homework = 0.45 \* (((Integer.parseInt(tokens[1]) + Integer.parseInt(tokens[2]) + Integer.parseInt(tokens[3])))/3);**

**double project = 0.25 \* Integer.parseInt(tokens[5]);**

**double midFinal = 0.30 \* (((Integer.parseInt(tokens[4]) + Integer.parseInt(tokens[6])))/2);**

**int final\_numeric\_grade = (int) (homework + project + midFinal);**

**//Requirement3:Determining the final letter grade**

**//calculate the letter grade and grade counts from A-F in 0-4 indices**

**char letter\_grade = 'F';**

**if(final\_numeric\_grade >= 90)**

**{**

**letter\_grade = 'A';**

**gradeCount[0]++;**

**}**

**else if(final\_numeric\_grade >= 80)**

**{**

**letter\_grade = 'B';**

**gradeCount[1]++;**

**}**

**else if(final\_numeric\_grade >= 70)**

**{**

**letter\_grade = 'C';**

**gradeCount[2]++;**

**}**

**else if(final\_numeric\_grade >= 60)**

**{**

**letter\_grade = 'D';**

**gradeCount[3]++;**

**}**

**else**

**{**

**gradeCount[4]++;**

**}**

**//Requirement5:GUI to display the final letter grade for each student**

**// add array consisting name and grade to data**

**gradeData.add(new String[] { tokens[0], letter\_grade + "" });**

}

**//Requirement4:GUI for display grades and counts**

**JFrame frame = new JFrame("Final Grades Calculator");**

**frame.setLayout(new BorderLayout());**

**JTable table = new JTable(gradeData.toArray(new String[0][0]), new String[] { "Name of Student", "Grade of Student" });**

**//Requirement4:Adding ScrollPane**

**JScrollPane jsp = new JScrollPane(table);**

frame.getContentPane().add(jsp);

**//Requirement5:Displaying Grade and its count**

**JLabel label = new JLabel("A:" + gradeCount[0] + " B:" + gradeCount[1] + " C:" + gradeCount[2] + " D:" + gradeCount[3] + " F:" + gradeCount[4], SwingConstants.CENTER);**

**frame.getContentPane().add(label, BorderLayout.PAGE\_END);**

frame.setSize(400, 400);

frame.setVisible(true);

frame.setResizable(false);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

}

catch (Exception e)

{

e.printStackTrace();

}

finally

{

if (bufferedReader != null)

{

try

{

bufferedReader.close();

}

catch (Exception e)

{

e.printStackTrace();

}

}

}

}

}

**Output:**





