Algorithms:

1. **Method int[] dimensionsFile(String filename):**

This method takes in a string (from the text file), returns an array, and is set up to read the text file. It first defines an array called dimensions that have only two dimesions (for number of lines and number of columns). Then it sets up and reads the text file. An integer counter is initialized. A while loop is used until there are no more lines to scan.

Within the loop, if else statement is used. If counter equal to zero, then a line from the text file will input the total amount of elements in the string into an array. It will repeat this until the loop is done. It then returns the array.

2. **Method int[][] readGradesFromFile(String filename):**

This method takes a string (from the text file), returns a 2D array, and is set up to read the text file. It first defines an array from dimensionsFile, and defines a 2D array called grades. The size of the 2D array with be from the values of the dimensionsFile. It then sets up and reads the text file. A new 2D array is defined (newGrades) to store the extracted values. An interger counter is initialized for later use. A nest for loop is then executed to read line by line and extract the numbers from the string.

An array of strings (currentLine) is initialized to get the elements extracted from the string lines. Counter is assigned to zero. The next for loop that reads each element of the string lines. An if statement is used to find numbers from the elements (makes sure if the element matches a number), and converts the element to a number and assigns it to a new array and adds one to counter (which is uses to assign values to the 2D array. It finally returns the new 2D array.

3. **Method double[] computeAverages(int[][] A):**

This method takes in 2D array and returns a double array. It first begins by creating a double array (called result) by adding the dimensions of the 2D array (from the last method). Then it runs a for loop, and runs with number of lines from the 2D array. Within the loop, the first array elements are assigned by the method averageAtGivenRow (method that get averages from the rows).

Then another for loop is used, and runs with the number of columns. Within the loop, the other array elements are assings by the method averageAtGivenColumn (method the get averages from the columns). The array is then finally returned.

4. **Method double averageAtGivenRow(int[][] A, int line):**

This method takes in a 2D array and an integer, and returns a double. It first begins by initializing a double called b to zero. Then a for loop is uses and runs the number of elements in the row. The double b is added with the array elements of the row. A new double is declared (called avgRow) is assigned by the division of the b and the number of elements in the row. Finally, avgRow is returned.

5. **Method double averageAtGivenColumn(int [][] A. int line):**

This method takes in a 2D array and an integer, and returns a double. It first begins by initializing a double called b to zero. Then a for loop is uses and runs the number of elements in the row. The double b is added with the array elements of the row. A new double is declared (called avCol) is assigned by the division of the b and the number of elements in the row. Finally, avgCol is returned.

6. **Method void writeMaxToFile(double[] B, string filename):**

This method takes a double array and string (the text file), and returns nothing. This method’s purpose is to find the max value of the averages collected in the double array,, and print the highest one in to a new file. It first begins by initialzing a double called maxValue and assigning it to zero. It then runs a for loop the number of times of elements in the arrray. It uses an if statement to check each invividual value to compare each other. Then the method sets up the text reader and prints the max value to a new text file.