```
// This program will input an undetermined number of student names
// and a number of grades for each student. The number of grades is
// given by the user. The grades are stored in an array.
// Two functions are called for each student.
// One function will give the numeric average of their grades.
// The other function will give a letter grade to that average.
// Grades are assigned on a 10 point spread.
// 90-100 A 80-89 B 70-79 C 60-69 D Below 60 F

// Connor Seemann

#include <iostream>
#include <iomanip>
#include <vector>

using namespace std;

const   int    MAXGRADE = 25;  // maximum number of grades per student

typedef float GradeType[MAXGRADE];      // one dimensional integer array data
 type

float findGradeAvg(GradeType, int, int);// finds grade average by taking array
 of
                                        // grades and number of grades as
                                         parameters

char findLetterGrade(float);            // finds letter grade from average
 given
                                        // to it as a parameter
int main()
{
    vector<string> names;
    int numOfGrades, counter;           // holds the number of grades
    GradeType grades;                   // grades defined as a one dimensional
     array
    float average;                      // holds the average of a student's
     grade
    char moreInput;                     // determines if there is more input
    counter = 1;                        // holds the amount of people in the
     lists

    cout << setprecision(2) << fixed << showpoint;
      // setting the returning floating points to be 2 fixed points after the
     decimal

    // Input the number of grades for each student
    cout << "Please input the number of grades each student will receive."
      // inputs for the max grade to get the length of the array|
         << endl << "This must be a number between 1 and " << MAXGRADE
           //                                                    |
```

```
     << " inclusive" << endl;
       // ------------------------------------------------------|
cin >> numOfGrades;
  // input for the amount of grades that will bve entered

while (numOfGrades > MAXGRADE || numOfGrades < 1)
{
    cout << "Please input the number of grades for each student." << endl
      // instructions for the input of how many students there will be for
     each student |
    << "This must be a number between 1 and " << MAXGRADE
      //
                  |
    << " inclusive\n";
      //
     ------------------------------------------------------------------
     ---------|
    cin >> numOfGrades;
      // Gets the input for the amount of grades for each student
}

// Input names and grades for each student
cout << "instructions:\nenter \"y\" if you want to input more students"
  // asks if the user wants to input more students into the program |
     << " any other character will stop the input" << endl;
       // ------------------------------------------------------------|
{
    int pos = 0;
    do
      // do while the user wants to enter more names into the program,
     continue
    {
        string firstname, lastname;
        cout << "Please input the first name of the student" << endl;
          // prompts for the first name input
        cin >> firstname;
          // gets the first name of the student
        cout << "Please input the last name of the student"  << endl;
          // prompts for the last name of the student
        cin >> lastname;
          // getting the input for the lastname
        names.push_back(firstname + " " + lastname);
          // combines the first name and lastname so you don't need two
         arrays
        for (int i = 0; i < numOfGrades; i++)
          // for loop for storing the grades into the array grades range
         is numOfGrades
        {
            cout << endl << "Please input a grade" << endl;
              // prompt for grade input
```

```
            cin >> grades[i+pos];
              // getting the grade input
        }
        pos += numOfGrades;
        cout << "Please input a y if you want to input more students"
          // asks if the user wants to continue and enter more students
            << " any other character will stop the input" << endl;
              // averages into the application.
        cin >> moreInput;
          // gets the more input to see if the user want's to contunue.

        if (moreInput == 'Y' || moreInput == 'y')
          // if the user wants to enter more names then it adds to the
         counter
            counter++;

    } while (moreInput == 'y' || moreInput == 'Y');
      // while check for the do while loop
}

for (int i = 0; i < counter; i++)
  //for loop for outputting grades entered
{
    int pos = i * numOfGrades;
      // correcting the positioning for the for loops
    cout << names[i] << endl;
      // main print statement for the grade output|

    for (int j = 0; j < numOfGrades; j++)
      //                                          |
    {
      //                                          |
        cout << setw(10) << grades[j + pos] << endl;
          // ----------------------------------------|
    }
    cout << endl;
      // enhances readability
}
  // end of main for

for (int i = 0; i < counter; i++)
  // for loop to output the average grade for each person
{
    int pos = i * numOfGrades;

    float average = findGradeAvg(grades, numOfGrades, pos);
      // defining the average while calling on the findGradeAverage
     function

    cout << names[i] << " has an average of "
      // out puts the average of the grade firstname lastname got as |
```

```
            << average << " which gives the letter grade of "
              // well as the letter grade they got as well                    |
            << findLetterGrade(average);
              // ------------------------------------------------------------|

        cout << endl;
          // enhancing readability
    }
    cout << endl;
      // endl to end cleanly
    return 0;
      // returning 0 to end the program without any errors
}
  // end of main()

//
 ------------------------------------------------------------------------------
 ---------------------------- |
//                                        function deffinitions begin
                                 |
//
 ------------------------------------------------------------------------------
 ---------------------------- |

//*********************************************************************
// findGradeAvg
//
// task:          This function finds the average of the
//                numbers stored in an array.
//
// data in:       an array of integer numbers
// data returned: the average of all numbers in the array
//
//*********************************************************************

float findGradeAvg(GradeType array, int numGrades, int pos)
{
    double average;                                    // declairing a return
     variable

    for(int i = 0; i < numGrades; i++)                 // for loop to process
     all of the grades
       average += array[pos+i];                        // adding the array
        ammount to the average

    return average/numGrades;                          // deviding by the
     amount of grades to get the average
}

//*********************************************************************
// findLetterGrade
```

```
//
// task:          This function finds the letter grade for the number
//                passed to it by the calling function
//
// data in:       a floating point number
// data returned: the grade (based on a 10 point spread) based on the
//                number passed to the function
//
//*************************************************************************

char findLetterGrade(float average)
{
 //|-------------------------------------------------------------|//
    if (average >= 90 && average <= 100)            //| comparing the average
     to see what grade the student deserves |//
        return 'A';                                 //|
                                                      |//
    else if (average >=80 && average < 90)          //|
                                                    |//
        return 'B';                                 //|
                                                      |//
    else if (average >= 70 && average < 80)         //|
                                                    |//
        return 'C';                                 //|
                                                      |//
    else if (average >= 60 && average < 70)         //|
                                                    |//
        return 'D';                                 //|
                                                      |//
    else if (average < 60)                          //|
                                                    |//
        return 'F';                                 //|
                                                      |//
    else                                            //|
                                                    |//
        return 'N';
         //|_____________________________________________________________|//

}

// END OF PROGRAM
```