```cpp
/**
 * This program is a review of some 1043 (CS 1) topics.
 *
 * Functions
 * Structs
 * Arrays of Structs
 *
 */
#include <iostream> // write to and read from stdin and stdout
#include <ctime>    // access system clock (we used for rand function)
#include <fstream> // read a write files
#include<iomanip> // Used to format output.
#include<string>

#define SIZE 10 // gives us a CONSTANT to use anywhere in our program \
                // we capitalize constants so we know they are not variables!

using namespace std; // so we don't have to put std:: in front of cin, cout.


/**
 * Structs are like creating your own data type.
 * We can use this struct to represent a student and thier grades.
 *
 * Remember, to declare a variable we use the following:
 *
 *      int x;
 *
 * To use our new struct we do the same:
 *
 *      Student S;
 *
 * Then we can load it with data like:
 *
 *      S.fname = 'Susan';
 *      S.lname = 'Sarandon';
 *      S.numGrades = 1;
 *      S.grades[0] = 88;
 *
 */
struct Student
{
    string fname;
    string lname;
    int numGrades;
    int grades[10];

}; // REMEMBER TO TRY AND FIGURE THIS OUT LATER!


/**
 * Function: loadClassList
```

```
 * Description:
 *      Reads a file with student data, and loads it into an array of Students.
 *
 * Params:
 *      Student *classlist  : array of Students
 *      string  filename    : name of file to process
 *
 * Returns:
 *      int - number of students read in.
 */

ofstream outfile;
int loadClassList(Student *classList, string fileName)
{
    ifstream infile;       // get a stream variable
    infile.open("input_data.txt"); // open the stream using our fileName param
    int i = 0;          // index (counter)

    if (!infile) // Error message if file could not be opened
    {
        cerr << "Error: file could not be opened" << '\n';
        system("pause");
        exit(1);
    }
    while (!infile.eof())// Loops until the end of file pointer is reached.
    {

        infile >> classList[i].fname >> classList[i].lname >> classList
          [i].numGrades;

        for (int n = 0; n < classList[i].numGrades; n++)
        {
            infile >> classList[i].grades[n];
        }




        // increment `i`
        i++;
    }



    // return sudent count
    return i;

}

/**
 * Function: printClassList
 *
```

```
104  * Description:
105  *      Prints an array of students to stdout
106  *
107  * Params:
108  *
109  *      Student *classList : array of structs (and the structs are `Students`)
110  *      int     classSize  : size of class (returned from `loadClassList` )
111  *
112  * Returns:
113  *
114  *      void
115  */
116
117
118 void printClassList(Student *classList, int classSize)
119 {
120     ofstream outfile;
121     outfile.open("student_output.txt");
122     outfile << " Name: Cykelle Semper. \n";
123     outfile << " Course: CMPS 1063 Data Structures, Fall 2019, Dr. Griffin.\n";
124     outfile << " Purpose: This program is a review of CMPS 1044 Topics, \n";
125     outfile << " Functions, Structs, and Arrays of Structs. \n\n";
126     outfile << "Students \n";
127     outfile << "========================== \n";
128     for (int i = 0; i < classSize; i++)
129     {
130         outfile << classList[i].fname << " "
131             //<< classList[i].numGrades
132             << classList[i].lname << ":  ";
133
134
135             for (int n = 0; n < classList[i].numGrades; n++)
136             {
137                 outfile << setw(2);
138                 outfile <<" "<< classList[i].grades[n];
139                 outfile << setw(2);
140             }
141
142             outfile << "\n";
143
144     }
145
146
147
148 }
149
150
151
152 /**
153  * Function: main
154  *
155  * Description:
```

```
156  *      Drives this example of structs and arrays
157  */
158 int main()
159 {
160
161     int A[SIZE];        // Array to hold students read from file
162     Student classList[100];
163     int classSize = 0;
164
165     // Call the loadClassList function to open a file and load an array
166     // of `Student` structs.
167     classSize = loadClassList(classList, "input_data.txt");
168
169     // Function returned how many lines (students) read in, lets print it.
170
171     //outfile << "Class Size: " << classSize << endl;
172
173     // Function to print out an array of students in a formatted way.
174     printClassList(classList, classSize);
175
176
177     return 0;
178 }
```