

```
/* Name:Kyle Davidson
Date: April 20, 2015
Section: 3
Assignment: #7
Due Date: April 22, 2015
About this project: This project will create a program that will display a summary screen of student s'
grades in specific classes given a text file. The assignment will help better understand file input and output, structures,
dynamic memory allocation, functions, and arrays
Assumptions: Assumes the names in the text file is 20 characters or under
All work below was performed by Kyle Davidson */
```

```
#include <iostream>
#include <iomanip>
#include <cstring>
#include <cctype>
#include <fstream>
```

```
using namespace std;
```

```
struct student
{
    char l_name[20], f_name[20]; //two character arrays for the first and last name of each student
    char subject, stud_grades;
    int test1, test2, final_exam; //used to calculate the grade of each students exam and weight of each exam
    double total_test;
};
```

```
void stud_grades(student *sgrades, int size); //function which assigns letter grades
void average (student *sgrades, int size); //function
```

on which averages all the tests

```
void grade_sum (student *sgrades, int size, ofstream &output); //function which prints out a grade summary for each student in every subject
```

```
int main()
{
    int i, size;
    ifstream input;
    ofstream output;
    char file[20] = {'\0'};

    do //do while loop allowing the user to type in the input file until it is correctly typed in
    {
        input.clear();
        cout << "Please enter the name of the input file.\n";
        cout << "Filename: ";
        cin >> file;

        input.open(file);
        if (!input)
            cout << "That is not a valid file. Try again!\n";
    } while (!input);

    do //do while loop allowing the user to enter in an output file until it is correctly entered
    {
        output.clear();
        cout << "Please enter the name of the output file.\n";
        cout << "Filename: ";
        cin >> file;

        output.open(file);
        if (!output)
            cout << "That is not a valid file. Try again!\n";
    } while (!output);
}
```

```

input >> size;
student *sgrades; //declaring a pointer
sgrades = new student[size]; //memory
input.ignore(); //skips space

for(i = 0; i < size; i++)
{
    input.getline(sgrades[i].l_name, 20, ',');
    input.getline(sgrades[i].f_name, 20, ',');

    input >> (sgrades[i].subject);
    input.ignore();

    input >> sgrades[i].test1;
    input.ignore();

    input >> sgrades[i].test2;
    input.ignore();

    input >> sgrades[i].final_exam;
    input.ignore();
}

    grade_sum(sgrades, size, output); //calling the
grade_sum function
    input.close(); //closes input file
    delete [] sgrades; //deletes memory used

    /*output << sgrades[i].l_name << ",";
output << sgrades[i].f_name << ",";
output << sgrades[i].subject << ",";
output << sgrades[i].test1 << ",";
output << sgrades[i].test2 << ",";
output << sgrades[i].final_exam << endl;*/

    return 0;
}

void stud_grades(student *sgrades, int size) //fun
ction which assigns grades to each student
{
    ofstream output;

```

```

int i;

for(i = 0; i < size; i++) //for loop with if s
tatement assigning grades based on numeric value
{
    if(sgrades[i].total_test < 60.0) //if stud
ents total test grade is below a 60 percent give t
hem an 'F'
    {
        sgrades[i].stud_grades = 'F';
    }
    else if(sgrades[i].total_test < 70.0) //if
students total test grade is below 70.0 percent g
ive them a 'D'
    {
        sgrades[i].stud_grades = 'D';
    }
    else if(sgrades[i].total_test < 80.0) //if
students total test grade is below 80.0 percent g
ive them a 'C'
    {
        sgrades[i].stud_grades = 'C';
    }
    else if(sgrades[i].total_test < 90.0) //if
students total test grade is below 90.0 percent g
ive them a 'B'
    {
        sgrades[i].stud_grades = 'B';
    }
    else //if students total test grade is bel
ow 100.0 percent, but above 90.0 percent give them
an 'A'
    {
        sgrades[i].stud_grades = 'A';
    }
}
}

```

```

void average (student *sgrades, int size) //void f
unction which calculates the averages/percentages
of each students tests

```

```

{
    int i;
    for(i = 0; i < size; i++) //for loop going through each students test grades and calculating the
    m
    {
        sgrades[i].total_test = (sgrades[i].test1*
0.3)+(sgrades[i].test2*0.3)+(sgrades[i].final_exam
*0.4); //formula for each students test average
    }
}

void grade_sum (student *sgrades, int size, ofstre
am &output) //void function which prints out all t
he students grades for each subject
{
    average (sgrades,size); //calling the test aver
age function
    stud_grades(sgrades,size); //calling the letter
grade function

    int english = 0, history = 0, math = 0;
    double english_tests = 0, history_tests = 0, ma
th_tests = 0;

    output << "Student Grade Summary" << endl;
    output << "-----\n";
    output << "\nENGLISH CLASS" << endl << "\nStude
nt Name" << "\t\t\t\t\tTest Avg" << endl;
    output << "-----
\n";
    for(int i = 0; i < size; i++) //for loop which
iterates through all the students and grades
    {
        if(sgrades[i].subject == 'E') //if stateme
nt that outputs each students first name and last
name based on each subject
        {
            output << sgrades[i].f_name;
            output << " " << sgrades[i].l_name;
            output << "\t\t\t\t\t";
            output << fixed << showpoint << setprec

```

```

ision(2);
        output << sgrades[i].total_test << '\t'
;
        output << sgrades[i].stud_grades << '\
n';
        english++; //incrementing students who
are in English class
        english_tests = english_tests + sgrades
[i].total_test;
    }

}
    output << "\n\t\tClass Average\t\t" << (englis
h_tests/english) << endl; //average for english te
sts in the class
    output << "-----
--\n\nHISTORY CLASS\n" << endl;
    output << "Student Name" << "\t\t\tTest Avg"
<< endl;
    output << "-----
--" << endl;

    for(int i = 0; i < size; i++) //for loop itera
ting through all the students and grades in Histor
y class
    {
        if(sgrades[i].subject == 'H') //if stateme
nt which outputs first and last name of students t
aking History class
        {
            output << sgrades[i].f_name;
            output << " " << sgrades[i].l_name;
            output << "\t\t\t";
            output << fixed << showpoint << setprec
ision(2);
            output << sgrades[i].total_test << '\t'
;
            output << sgrades[i].stud_grades << '\
n';
            history++; //incrementing students who
are in History class

```

```

        history_tests = history_tests + sgrades
[i].total_test;
    }
}

    output << "\n\t\tClass Average\t\t" << (histor
y_tests/history) << endl; //class average for hist
ory tests
    output << "-----"
--\n\nMATH CLASS\n" << endl;
    output << "Student Name" << "\t\t\tTest Avg"
<< endl;
    output << "-----"
--" << endl;

    for(int i = 0; i < size; i++) //for loop itera
ting through every student and grade in Math class
    {
        if(sgrades[i].subject == 'M') //if stateme
nt which outputs first and last name of students a
nd grades in math class
        {
            output << sgrades[i].f_name;
            output << " " << sgrades[i].l_name;
            output << "\t\t\t";
            output << fixed << showpoint << setprec
ision(2);
            output << sgrades[i].total_test << '\t'
;
            output << sgrades[i].stud_grades << '\
n';
            math++; //incrementing students who are
in math class
            math_tests = math_tests + sgrades[i].to
tal_test;
        }
    }
}

```

```
    output << "\n\t\tClass Average\t\t" << (math_t  
ests/math) << endl; //class test average for stude  
nts in math class  
    output << "-----  
--" << endl;
```

```
}
```