Nick Krisulevicz

Dr. Wang

COSC 120-751

10/30/2020

Project 2

**Part 1: Algorithm**

1. The project is intended to take individual grades from a file, and calculate the final grades for each student. Each student has their grade data on one line in the file. The program will open the file, read the grades, perform the calculations, and close the file. The final grade calculation will be output at the end.
2. The arrays we will need for this project are: quiz size, lab size, project size, and exam size. Quiz size will have an int data type, be named quiz[], have 10 elements, and be used for storing the quiz grades. Lab size will have an int data type, be named lab[], have 10 elements and be used for storing the lab grades. Project size will have an int data type, be named project[], have 3 elements and be used for storing the project grades. Exam size will have an int data type, be named exam[], have three elements and be used for storing exam grades.
3. The functions we will need for the project are: get average, calculate grade, get letter, and get penalty. Get average will be named getaverage(), have a float return type, have int total and int size as its formal parameters. Its purpose is to calculate initial averages for each category, quiz, lab, project, and midterm. Calculate grade will be named calculategrade(), have float as the return type, have float avg1, float avg2, float avg3, float avg4 as its formal parameters. Its purpose is to calculate a total average grade from each of the four initial averages. Get penalty will be named getpenalty(), have a float return type, and have float average as its formal parameter. The purpose is to calculate the final grade with the attendance penalty included in the grade. The last function is get letter, and it will be named getletter(), have char return type, will have float totalavg, float projectavg as its formal parameters. Its purpose is to assign a letter grade to each student’s grade based off their total average with the added condition of the project average being a certain amount. The function prototypes are as follows:

float getaverage(int total, int size);

float calculategrade(float avg1, float avg2, float avg3, float avg4);

char getletter(float totalavg, float projectavg);

float getpenalty(float average);

1. Pseudocode for the project is an overview of what the program does. First, the program opens the external file grade120.dat. It outputs a message saying if the file opened or not. Next, the program reads in the first name, last name, each of the four grades, the attendance, and the final grade of each student. Next, the program uses functions to calculate the final grade of each student given the conditions to get their grade. Next, the program outputs the final grades to the console and writes them in a separate file called letter120.dat

**Part 2: Code**

The source code is as follows:

//Nick Krisulevicz

//Project 2

//10/30/2020

#include <iostream>

#include <fstream>

#include <iomanip>

#include <string>

using namespace std;

//constants

const int QUIZSIZE = 10;

const int LABSIZE = 10;

const int PROJSIZE = 3;

const int EXAMSIZE = 3;

const int STUDENTS = 19;

const double ATTENDANCEPENALTY = 0.01;

//function prototypes

float getaverage(int total, int size);

float calculategrade(float avg1, float avg2, float avg3, float avg4);

char getletter(float totalavg, float projectavg);

float getpenalty(float average);

//main function

int main()

{

//variables

ifstream datain;

ofstream dataout;

string heading;

string firstname;

string lastname;

int quiz[QUIZSIZE];

int lab[LABSIZE];

int project[PROJSIZE];

int exam[EXAMSIZE];

double attendance;

double finalexam;

int counter = 1;

int total;

int tempsize;

float quizaverage = 0;

float labaverage = 0;

float projectaverage = 0;

float examaverage = 0;

float totalaverage = 0;

float finalaverage = 0;

char lettergrade;

//opening the file

dataout.open("letter120.dat");

datain.open("grade120.dat");

if (datain)

{

cout << "File found" << endl;

}

else

{

cout << "File not found" << endl;

}

//reading the grades from the file

while (!datain.eof())

{

if (counter == 1 || counter == 2) //gets the first two lines as the heading

{

getline(datain, heading);

cout << heading << endl;

}

else if (counter < 22){ //stops reading the file after it has retrieved all the lines

break;

}

else //The process of reading the file contents

{

datain >> firstname;

datain >> lastname;

for (int i = 0; i < QUIZSIZE; i++) //puts quiz grades into array

{

datain >> quiz[i];

total = total + quiz[i];

cout << quiz[i];

}

tempsize = QUIZSIZE;

quizaverage = getaverage(total, tempsize);

total = 0;

for (int i = 0; i < LABSIZE; i++) //puts lab grades into array

{

datain >> lab[i];

total = total + lab[i];

cout << lab[i];

}

tempsize = LABSIZE;

labaverage = getaverage(total, tempsize);

total = 0;

for (int i = 0; i < PROJSIZE; i++) //puts project grades into array

{

datain >> project[i];

cout << project[i];

}

tempsize = PROJSIZE;

projectaverage = getaverage(total, tempsize);

total = 0;

for (int i = 0; i < EXAMSIZE; i++) //puts exam grades into array

{

datain >> exam[i];

cout << exam[i];

}

tempsize = EXAMSIZE;

examaverage = getaverage(total, tempsize);

datain >> finalexam;

datain >> attendance;

counter++;

for (int i = 0; i < STUDENTS; i++) //calculates average grade for each student

{

totalaverage = calculategrade(quizaverage, labaverage, projectaverage, examaverage);

finalaverage = getpenalty(totalaverage); //uses getpenalty function to adjust final grade

cout << "Final average is " << finalaverage << endl;

}

}

}

counter = 1;

while (!datain.eof()) //writes data to letter120.dat

{

if (counter == 1 || counter == 2)

{

getline(datain, heading);

dataout << heading;

counter++;

}

else if (counter < 22)

{

break;

}

else{

dataout << firstname;

dataout << lastname;

lettergrade = getletter(finalaverage, projectaverage);

dataout << lettergrade;

}

}

datain.close(); //closes grade120.dat file

dataout.close(); //closes letter120.dat file

return 0;

}

//getaverage function definition

float getaverage(int total, int size)

{

return static\_cast<float>(total/size);

}

//calculategrade function definition

float calculategrade(float avg1, float avg2, float avg3, float avg4)

{

float totalavg = avg1 + avg2 + avg3 + avg4;

float finalavg = totalavg / 4;

return finalavg;

}

//getletter function definition

char getletter(float totalavg, float projectavg)

{

char letter;

if (totalavg >= 90.0 && projectavg >= 90.0)

{

letter = 'A';

}

else if (totalavg >= 90.0 && totalavg < 80.0 && projectavg >=80.0)

{

letter = 'B';

}

else if(totalavg >= 80.0 && totalavg > 70.0 && projectavg >= 70.0)

{

letter = 'C';

}

else

{

letter = 'F';

}

return letter;

}

float getpenalty(float average)

{

float penalty = average \* ATTENDANCEPENALTY;

float grade = average - penalty;

return grade;

}

**Part 3: Testing**

The console output displayed the contents of grade120.dat just fine. The program also created letter120.dat when the program executed. However, the contents of letter120.dat would not write, and I could not figure out how to get it to write. The output the program finished with is as follows:

File found

First Last Q0 Q1 Q2 Q3 Q4 Q5 Q6 Q7 Q8 Q9 L0 L1 L2 L3 L4 L5 L6 L7 L8 L9 P0 P1 P2 E0 E1 E2 FI ATT

------------------------------------------------------------------------------------------------------------------------------------------

Kevin Smith 90 100 100 100 98 97 87 100 85 87 89 100 100 100 100 90 100 98 90 100 98 98 98 90 90 98 88 0.00

Morgan Kelly 80 100 65 67 69 71 100 100 100 67 95 85 87 89 100 65 67 69 71 100 98 98 98 65 67 69 71 0.10

Isaac Newton 100 90 100 90 100 90 100 90 100 100 100 100 100 100 100 100 100 100 100 100 98 98 98 90 90 98 88 0.00

Cole Jones 100 100 100 87 73 75 77 79 81 87 89 91 73 75 77 79 81 100 100 100 98 100 65 67 69 71 63 0.05

Angela Allen 100 100 100 87 89 91 93 95 100 100 100 100 100 100 100 95 97 100 98 98 98 90 73 75 77 79 81 0.02

David Cooper 56 58 60 62 64 100 100 100 87 73 75 77 100 100 77 79 81 100 100 100 98 70 72 74 76 78 88 0.00

Nancy Bailey 100 87 89 91 93 95 100 100 100 100 100 100 91 93 95 97 100 98 100 100 98 98 98 90 90 98 88 0.00

Emily Synder 65 67 69 71 62 64 73 75 77 58 60 62 79 66 68 70 72 81 74 76 78 90 90 74 76 98 88 0.00

Lori Austin 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 98 98 98 90 90 98 88 0.02

Jenny Howard 56 58 60 62 71 62 64 73 100 66 68 70 72 74 76 78 100 100 100 60 62 79 66 68 70 68 70 0.03

Anne Lewis 100 86 58 60 100 71 62 64 73 94 66 68 90 72 74 76 78 67 68 69 70 71 98 88 76 78 68 0.04

Nick Johnson 100 100 89 91 73 75 77 79 81 100 100 100 98 100 100 95 85 87 89 100 98 98 98 80 76 78 98 0.01

Nick Spickler 100 93 95 97 100 98 98 98 90 100 89 91 93 95 97 100 100 89 91 93 95 97 98 98 90 90 98 0.00

Joy Williams 75 77 58 60 62 79 66 68 70 72 81 100 100 71 62 64 73 94 66 98 90 90 98 68 90 88 77 0.00

Barbara Hood 100 67 95 85 87 89 91 93 95 97 85 87 100 100 100 71 62 64 73 94 66 68 98 98 90 90 88 0.00

Joe Hoarn 62 63 64 65 66 67 68 69 70 71 100 81 100 100 71 62 64 73 100 100 98 98 64 73 94 66 68 0.08

Payton Bardzell 100 100 100 97 87 67 95 85 87 89 91 93 95 97 100 100 100 95 85 87 89 100 98 90 90 78 98 0.00

Kim Ludwig 71 62 64 73 75 77 58 60 62 79 66 68 70 72 81 100 100 79 66 68 70 72 98 98 90 90 98 0.09

Susan Honks 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 90 90 88 100 100 100 100 0.00

Letter120.dat has no contents

In terms of help, Ashwin and Troy both asked me for help, and I could not give them an answer because I was struggling myself. I texted Cole and asked for help getting the program to write to letter120.dat, but he did not respond.