Program 3

#include <iostream>

#include <string>

#include <vector>

#include <fstream>

using namespace std;

struct Students

{

long long id;

string first;

string last;

string answers;

int score;

float percent;

string grade;

};

string StringToUpper(string str)

{

for (int i = 0; i < str.length(); i++)

{

str[i] = toupper(str[i]);

}

return str;

}

void FillVector(vector<Students>& list, ifstream& pull)

{

Students temp;

while (!pull.eof())

{

pull >> temp.id;

pull >> temp.first;

pull >> temp.last;

pull >> temp.answers;

temp.answers = StringToUpper(temp.answers);

list.push\_back(temp);

}

}

void NumCorrect(vector<Students>& list, int numQuestions, string key)

{

int numCorrect;

for (int i = 0; i < list.size(); i++)

{

numCorrect = 0;

for (int j = 0; j < numQuestions; j++)

{

if (list.at(i).answers[j] == key[j])

{

numCorrect++;

}

}

list.at(i).score = numCorrect;

}

return;

}

void CalculatePercent(vector<Students>& list, int NumQuestions)

{

for (int i = 0; i < list.size(); i++)

{

list.at(i).percent = (static\_cast<float> (list.at(i).score) / NumQuestions) \* 100.0;

}

}

void CalculateGrade(vector<Students>& list)

{

for (int i = 0; i < list.size(); i++)

{

if (list.at(i).percent >= 90)

{

list.at(i).grade = "A";

}

if (list.at(i).percent < 90 && list.at(i).percent >= 80)

{

list.at(i).grade = "B";

}

if (list.at(i).percent < 80 && list.at(i).percent >= 70)

{

list.at(i).grade = "C";

}

if (list.at(i).percent < 70 && list.at(i).percent >= 60)

{

list.at(i).grade = "D";

}

if (list.at(i).percent < 60 && list.at(i).percent >= 50)

{

list.at(i).grade = "E";

}

if (list.at(i).percent < 50)

{

list.at(i).grade = "F";

}

}

}

void OutPutGrades(vector<Students>& list, ofstream& push)

{

for (int i = 0; i < list.size(); i++)

{

push << list.at(i).id << " ";

push << list.at(i).first << " ";

push << list.at(i).last << " ";

push << list.at(i).percent << "% ";

push << list.at(i).grade << endl;

}

}

void FindLowestAndHigestScore(vector<Students>& list, int& highpos, int& lowpos)

{

highpos = 0;

lowpos = 0;

int high = list.at(0).score;

int low = list.at(0).score;

for (int i = 0; i < list.size(); i++)

{

if (list.at(i).score > list.at(highpos).score)

{

highpos = i;

}

if (list.at(i).score < list.at(lowpos).score)

{

lowpos = i;

}

}

}

double FindAverage(vector<Students>& list)

{

int i = 0;

double avg = 0.0;

for (i = 0; i < list.size(); i++)

{

avg += list.at(i).score;

}

return avg / i;

}

string ClassGrade(double avg, int numQuestions)

{

double percent = avg / numQuestions \* 100.0;

if (percent >= 90)

{

return "A";

}

if (percent < 90 && percent >= 80)

{

return "B";

}

if (percent < 80 && percent >= 70)

{

return "C";

}

if (percent < 70 && percent >= 60)

{

return "D";

}

if (percent < 60 && percent >= 50)

{

return "E";

}

if (percent < 50)

{

return "F";

}

}

int main()

{

string userFile;

int lowpos = 0;

int highpos = 0;

double avg;

string grade;

vector<Students> list;

ifstream pull;

int numQuestions;

string key;

string examfile;

ofstream push;

cout << "Please enter the answer file name ";

cin >> userFile;

pull.open(userFile);

push.open("classgrades.txt");

pull >> numQuestions;

pull >> key;

pull >> examfile;

pull.close();

pull.open(examfile);

FillVector(list, pull);

NumCorrect(list, numQuestions, key);

CalculatePercent(list, numQuestions);

CalculateGrade(list);

OutPutGrades(list, push);

FindLowestAndHigestScore(list, highpos, lowpos);

cout << "The high score was " << list.at(highpos).score << " by " << list.at(highpos).first << " " << list.at(highpos).last << " " << list.at(highpos).id << endl;

cout << "The lowest score was " << list.at(lowpos).score << " by " << list.at(lowpos).first << " " << list.at(lowpos).last << " " << list.at(lowpos).id << endl;

avg = FindAverage(list);

cout << "The class average score was " << avg << endl;

grade = ClassGrade(avg, numQuestions);

cout << "Class average grade was " << grade << endl;

cout << "The number of students processed was " << list.size() << endl;

push.close();

push.open("statistics.txt");

push << "The high score was " << list.at(highpos).score << " by " << list.at(highpos).first << " " << list.at(highpos).last << " " << list.at(highpos).id << endl;

push << "The lowest score was " << list.at(lowpos).score << " by " << list.at(lowpos).first << " " << list.at(lowpos).last << " " << list.at(lowpos).id << endl;

push << "The class average score was " << avg << endl;

push << "The number of students processed was " << list.size() << endl;

return 0;

}









