**Final copy**

/\*This is a program for a student database. The user will need

to enter the students ID number as well as their names before they

can add any courses. The student list is sorted by the students ID

number due to the fact there will never be a duplicate just like a

social security number. The Students ID number is the KEY. Have fun

with this program. if you have any questions or need support please

email at walter.geanacopoulos@qc.quincycollege.edu\*/

/\*Creator: Walter Geanacopoulos

Due Date: 22 Dec 2016

Task: Final Progect\*/

#include <iostream>

#include <string>

#include <vector>

#include <fstream>

#include <algorithm>

#include <ctype.h>

using namespace std;

// class named course is where all the course info can be located

class course

{

private:

string name;

int num\_credits;

double grade;

public:

// default constructor for course

course()

{

}

// course name , credits and grade based off percentage

course(string name, int num\_credits, double grade)

{

this->name = name;

this->num\_credits = num\_credits;

this->grade = grade;

}

// function for get course name

string getName()

{

return name;

}

//function to print course info format

void print()

{

cout << "Course Name: " << name;

cout << "\nNum Credits: " << num\_credits;

cout << "\nGrade by percent: " << grade << "%" << endl;

}

};

// class for node

class node

{

public:

course data;

node \*next;

node(course data)

{

this->data = data;

next = NULL;

}

};

//class for student

class student

{

private:

string fName;

string lName;

int idNumber;

node \*head;

//public student info first name, last name and id number of student

public:

student(int idNumber, string fName, string lName)

{

this->fName = fName;

this->lName = lName;

this->idNumber = idNumber;

head = NULL;

}

//function to get name of student

string getName()

{

return lName, fName;

}

//function to get id number of student

double getidNumber()

{

return idNumber;

}

//function to add a course for the student

void add\_course(course c)

{

node \*newNode = new node(c);

newNode->next = head;

head = newNode;

}

//function to delete a course for a student format

void remove\_course(string lName)

{

if (head != NULL)

{

if (head->data.getName() == lName)

{

head = head->next;

cout << "course sucessfully removed" << endl;

return;

}

else

{

node \*temp = head;

if (temp->next != NULL)

{

if (temp->next->data.getName() == lName)

{

temp->next = temp->next->next;

cout << "course sucessfully removed" << endl;

return;

}

temp = temp->next;

}

}

}

cout << "course not found" << endl;

}

//function to print a students info format

void print()

{

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";

cout << "ID Number: " << idNumber << endl;

cout << "Last Name: " << lName << endl;

cout << "First Name: " << fName << endl;

cout << "--------Courses--------- " << endl;

node \*temp = head;

while (temp != NULL)

{

temp->data.print();

temp = temp->next;

}

}

};

//function to find a student by its id number

int find\_student(vector<student> students, int idNumber)

{

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

{

if (students[i].getidNumber() == idNumber)

{

return i;

}

}

return -1;

}

//function to add a student

void add\_student(vector<student> &students)

{

int idNumber;

string fName;

string lName;

cout << "Enter Students ID Number: ";

cin >> idNumber;

cout << "Enter Students First Name: ";

cin >> fName;

cout << "Enter Students Last Name: ";

cin >> lName;

transform(fName.begin(), fName.end(), fName.begin(), ::toupper);

transform(lName.begin(), lName.end(), lName.begin(), ::toupper);

students.push\_back(student(idNumber, fName, lName));

}

//function to delete a student

void remove\_student(vector<student> &students)

{

int idNumber;

cout << "Enter ID number of the student to remove: ";

cin >> idNumber;

int index = find\_student(students, idNumber);

if (index == -1)

{

cout << "Student not found" << endl;

}

else

{

for (int i = index; i < students.size() - 1; ++i)

{

students[i] = students[i + 1];

}

students.pop\_back();

}

}

//function to call the individual students info

void print\_student(vector<student> students)

{

int idNumber;

cout << "Enter ID number of the student you want to display: ";

cin >> idNumber;

int index = find\_student(students, idNumber);

if (index == -1)

{

cout << "Student not found" << endl;

}

else

{

student s = students[index];

s.print();

}

}

//function to sort the students by the ID number

void sort\_student(vector<student> &students)

{

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

{

for (int j = i + 1; j < students.size(); j++)

{

if (students[i].getidNumber() > students[j].getidNumber())

{

student temp = students[i];

students[i] = students[j];

students[j] = temp;

}

}

}

}

//function to print the entire database on the students

void print\_studentList(vector<student> &students)

{

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

{

students[i].print();

}

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";

}

//function to add a course to a specific student

void add\_course(vector<student> &students)

{

int idNumber;

cout << "Enter ID number of the student to add a course: ";

cin >> idNumber;

int index = find\_student(students, idNumber);

if (index == -1)

{

cout << "Student not found" << endl;

}

else

{

string name;

int num\_credits;

double grade;

cout << "Enter Course Name: ";

cin >> name;

transform(name.begin(), name.end(), name.begin(), ::toupper);

cout << "Number of Credits: ";

cin >> num\_credits;

cout << "Grade: ";

cin >> grade;

students[index].add\_course(course(name, num\_credits, grade));

}

}

// function to remove a course from a selected student

void remove\_course(vector<student> &students)

{

int idNumber;

cout << "Enter ID number of the student to remove a course: ";

cin >> idNumber;

int index = find\_student(students, idNumber);

if (index == -1)

{

cout << "Student not found" << endl;

}

else

{

string name;

cout << "Enter course name: ";

cin >> name;

students[index].remove\_course(name);

}

}

// function to displays the menu for the database

int get\_menu\_choice()

{

cout << "1. Add Student" << endl;

cout << "2. Remove Student" << endl;

cout << "3. Add Course" << endl;

cout << "4. Remove Course" << endl;

cout << "5. Display Individual Student's Details" << endl;

cout << "6. Display Sorted Student List By ID Number " << endl;

cout << "7. Exit" << endl;

int choice;

while (true)

{

cout << "Enter Your Choice: ";

cin >> choice;

cout << "\n";

if (choice < 1 || choice > 7)

{

cout << "Invalid Choice. Try Again!!" << endl;

}

else

{

return choice;

}

}

}

// main function where all the function calls come from as well as the

// menu choise case statements

int main()

{

vector<student> students;

int choice;

while (true)

{

choice = get\_menu\_choice();

switch (choice)

{

case 1:

add\_student(students);

break;

case 2:

remove\_student(students);

break;

case 3:

add\_course(students);

break;

case 4:

remove\_course(students);

break;

case 5:

print\_student(students);

break;

case 6:

sort\_student(students);

print\_studentList(students);

break;

case 7:

return 0;

}

cout << endl << endl;

}

return 0;

}



