|  |
| --- |
| ~ |
| Program 8: Student Class |
| Driven by a basic menu |
|  |
| **Brett Kriz** |
| **4/15/2014** |

|  |
| --- |
| ~ |

Prog8.cpp:

/\*------------------------------------------

Brett Kriz

Program 8

-------------------------------------------\*/

#include <algorithm> // Algorithms >.>

#include <cmath> // Basic Math Operations

#include <cstdlib> // control of stdlib

#include <cstring> // Cstrings

#include <ctime> // Time stuff

#include <ctype.h> // typeing

#include <fstream> // File stream (I/O)

#include <iomanip> // Manipulation of I/O

#include <iterator>

#include <iostream> // Input\Output

#include <sstream> // String stream

#include <stdlib.h> // Standard Library

#include <string> // String Stuff

#include <string.h> // String Stuff

#include <time.h> // More time stuff

#include "Student.h"

// SEE HEADER

using namespace std;

void body();

void sleep(unsigned int mseconds = 250);

void printall(Student s[], int size);

int upd(int& n);

int main()

{

system("color F0");

system("title PROGRAM 8: Student Class");

body();

cout << "\n";

system("pause");

return 0;

}

// Code Body

void body(){

int x = 0, size = 0;

bool go = true;

Student all[50];

// Create a table of students, then add

all[upd(size)] = Student("Starter Student","1010", 75,85,95,1);

all[upd(size)] = Student("Stu Little", "8008", 45,55,65,0);

all[upd(size)] = Student("AJ", "125", 25,25,25,1);

all[upd(size)] = Student("Claude Speed", "2001", 25,75,75,3);

all[upd(size)] = Student("Alice Wounder", "165", 25,25,25,135);

all[upd(size)] = Student("Erins Neverdone", "205", 35,25,11,0);

all[upd(size)] = Student("Brett K", "305", 25,25,25,8);

all[upd(size)] = Student("BURR SWAG", "40585", 25,50,25,10);

all[upd(size)] = Student("Fox Bob", "507", 25,25,25,20);

all[upd(size)] = Student("Klondike Smith", "555", 98,95,66,100);

all[upd(size)] = Student("Ender Student ~ SEE BELOW FOR ADDED", "5075", 25,25,25,100);

string arg = "";

do{ // LOOP for a menu

cout << "\nWould you like to (add,view,exit)?: ";

cin >> arg;

cin.clear();

if (arg.find("add") < arg.npos){ // npos for not found

Student z(true); // tag it for client

all[size] = z;

size++;

sleep();

//size++;

}else if(arg.find("view")<arg.npos){

system("cls");

printall(all,size); // PRINT ALL RECORDS

sleep(1000);

}else if(arg.find("exit")<arg.npos){

go = false; // Itll just stop this loop

}else{

cout << "\n\n\t What? I dont know what that means!\n"; // bad in

sleep();

}

}while(go); // end menu

}

void printall(Student s[], int size){

int x = 0;

while(x < size){

s[x].Print(); // Print record

x++;

sleep(); // Fo decoration

}

}

void sleep(unsigned int mseconds){

// Sleep...

clock\_t goal = mseconds + clock();

while (goal > clock());

}

int upd(int& n){

int nn = n;

n++;

return nn;

}

Student.h:

#include <cmath>

#include <iostream>

#include <iomanip>

#include <string>

#include <string.h>

#include <sstream>

using namespace std;

bool doover(bool b); // 1 proto

class Student{

private:

string name, ID;

int score1, score2, score3, absences;

public:

Student();

Student(string n, string id, int s1, int s2, int s3, int a);

Student(bool hi); // Just to change how its stacked

void Print();

string getName();

string getID();

int getScore1();

int getScore2();

int getScore3();

int getAbsences();

void setName();

void setID();

void setScore1();

void setScore2();

void setScore3();

void setAbsences();

double getFinal();

string gageSuccess(double f);

};

// Constructor

Student::Student(bool hi){ // Menu creation style

setName();

setID();

setScore1();

setScore2();

setScore3();

setAbsences();

}

Student::Student(){

name = "";

ID = "";

score1 = 0;

score2 = 0;

score3 = 0;

absences = 0;

}

Student::Student(string n, string id, int s1, int s2, int s3, int a){

name = n;

ID = id;

score1 = s1;

score2 = s2;

score3 = s3;

absences = a;

}

void Student::Print(){

cout << "\nName: " << getName()

<< "\nID: " << getID()

<< "\nScore1: " << getScore1()

<< "\nScore2: " << getScore2()

<< "\nScore3: " << getScore3()

<< "\nAbsences: " << getAbsences()

<< "\nFinal Grade: "<< fixed << setprecision(2) << getFinal() << "%"

<< endl <<gageSuccess(getFinal())

<< endl;

}

// GETS AND SETS

void Student::setName(){

cin.clear();

string s = "";

cout << "\nPlease enter First Name: ";

cin >> s;

name = s;

cout << "\nPlease enter Last Name: ";

cin >> s;

name += " " + s;

cout << endl;

}

void Student::setID(){

bool bad = true, nondigits = true, tots = false;

string s = "";

cin.clear();

do{

bad = true;

nondigits = false;

cout << "\nPlease enter ID: ";

cin.ignore(25,'\n');

//cin.ignore(25, ' ');

cin >> s;

// do checks

int x =0;

while( x < s.size() ){

if (!isdigit(s[x])){

cout << s.substr(0,x) << endl;

nondigits = true;

break;

}

x++;

} // all digits

if (nondigits == false) {

int arg = atoi(s.c\_str());

bad = (arg < 100 || arg > 100000);

}

tots = bad || nondigits;

}while( doover(tots) );

ID = s;

cout << endl;

}

void Student::setScore1(){

int n = -1;

cin.clear();

do{

cout << "\nPlease enter Score1: ";

cin >> n;

}while( doover(n<0 || n>100) );

score1= n;

cout << endl;

}

void Student::setScore2(){

int n = -1;

cin.clear();

do{

cout << "\nPlease enter Score2: ";

cin >> n;

}while( doover(n<0 || n>100) );

score2 = n;

cout << endl;

}

void Student::setScore3(){

int n = -1;

cin.clear();

do{

cout << "\nPlease enter Score3: ";

cin >> n;

}while( doover(n<0 || n>100) );

score3 = n;

cout << endl;

}

void Student::setAbsences(){

int n = -1;

cin.clear();

do{

cout << "\nPlease enter Absences: ";

cin >> n;

}while( doover(n < 0 || n > 365\*4) );

absences = n;

cout << endl;

}

string Student::getName(){return name;}

string Student::getID(){return ID;}

int Student::getScore1(){return score1;}

int Student::getScore2(){return score2;}

int Student::getScore3(){return score3;}

int Student::getAbsences(){return absences;}

// OTHER

double Student::getFinal(){

return (score1+score2+score3)/3.00;

}

string Student::gageSuccess(double f){

if (f >= 73){

return "Successful";

}else{

return "Unsuccessful";

}

}

bool doover(bool b){

if (b){ // For while loops

system("cls");

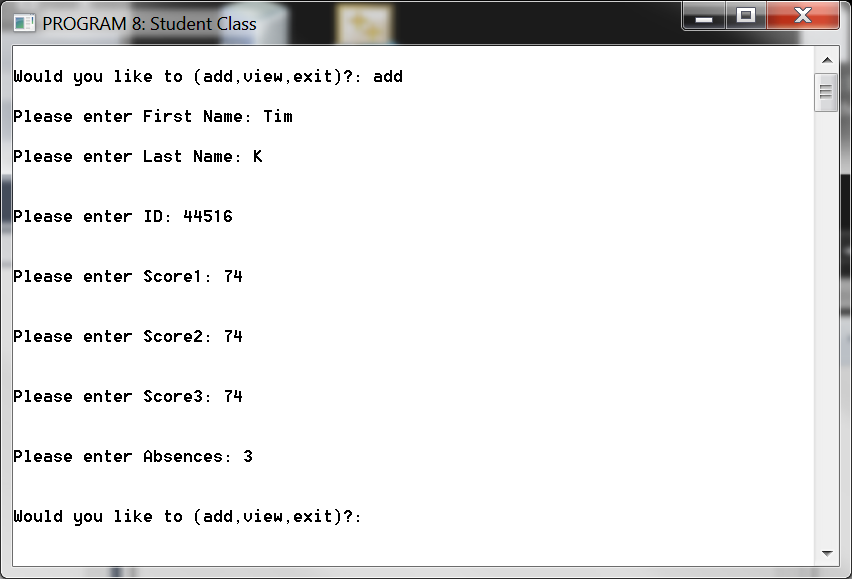
cout << "\nBad Input! Please re-enter!\n";

cin.clear();

}

return b;

}



Output:

