Source.cpp

#include<iostream>

#include"student.h"

using namespace std;

int main()

{

int a;

cin >> a;

course c1(1234, "English", a);

c1.enroll(55596, "Anas Zakir");

c1.enroll(47374, "Abdul Ahad");

c1.enroll(53849, "Ameer Muawiyah");

c1.enroll(53015, "Awais Ali");

c1.enroll(51599, "Tayyab Ayub");

c1.display();

cout << endl << endl;

c1.enroll(54378, "Faizan Ali");

c1.display();

c1.drop(53849);

c1.display();

c1.drop(924);

return 0;

}

Main.cpp

#pragma once

#include"student.h"

#include<iostream>

#include<string>

using namespace std;

//.........................Student imp.........................................

student::student()

{

id = -1;

name = "";

}

student::student( string name, int id)

{

this->id = id;

this->name = name;

}

void student::setId(int id)

{

this->id = id;

}

void student::setName(string name)

{

this->name = name;

}

int student::getId()

{

return id;

}

string student::getName()

{

return name;

}

//............................Course...........................

course::course()

{

courseId = 0;

courseName = "";

capacity = 20;

currentlyEnrolled = 0;

students = new student[capacity];

}

course::course(int id, string name)

{

courseId = id;

courseName = name;

capacity = 10;

currentlyEnrolled = 0;

students = new student[capacity];

}

course::course(int id, string name, int capacity)

{

courseId = id;

courseName = name;

this->capacity = capacity;

currentlyEnrolled = 0;

students = new student[capacity];

}

// getter and setter

void course::enroll(int id, string name)

{

if (currentlyEnrolled == capacity)

{

resize();

students[currentlyEnrolled].setId(id);

students[currentlyEnrolled].setName(name);

currentlyEnrolled++;

}

else

{

students[currentlyEnrolled].setId(id);

students[currentlyEnrolled].setName(name);

currentlyEnrolled++;

}

}

void course::resize()

{

capacity = capacity + 5;

student\* newStudents = new student[capacity];

for (int i = 0; i < currentlyEnrolled; i++)

{

newStudents[i].setId(students[i].getId());

newStudents[i].setName(students[i].getName());

}

delete[] students;

students = newStudents;

}

int course::search(int id)

{

for (int i = 0; i < currentlyEnrolled; i++)

{

if (students[i].getId() == id)

{

return i;

}

else

continue;

} //end of for

return -1;

}

void course::drop(int toDrop)

{

int index = search(toDrop);

if (index == -1)

{

cout << "Not found" << endl;

return;

}

else

{

students[index].setId(0);

students[index].setName("");

shift(index);

currentlyEnrolled--;

}

}

void course::shift(int index)

{

for (int i = index; i < currentlyEnrolled; i++)

{

students[i].setId(students[i + 1].getId());

students[i].setName(students[i + 1].getName());

}

}

void course::display()

{

cout << "\t\tCourse ID: " << courseId << endl;

cout << "\t\tCourse Name: " << courseName << endl;

cout << "\t\tTotal enrolled: " << currentlyEnrolled << endl;

cout << "\t\tRemaining slots: " << capacity - currentlyEnrolled << endl << endl;

for (int i = 0; i < currentlyEnrolled; i++)

{

cout << "\tStudent ID: " << students[i].getId() << " \t Student Name: " << students[i].getName() << endl;

}

}

Student.h

#include<iostream>

using namespace std;

class student

{

private:

string name;

int id;

public:

student();

student(string, int);

student(student&);

string getName();

void setName(string);

int getId();

void setId(int);

};

class course

{

private:

int courseId;

string courseName;

int capacity;

int currentlyEnrolled;

student\* students;

public:

course();

course(int, string);

course(int, string, int);

int getCourseId();

void setCourseId(int);

string getCourseName();

void setCourseName(string);

int getCapacity();

void setCapacity(int);

void enroll(int, string);

void drop(int);

void resize();

void shift(int);

int search(int);

void display();

};

Output: