Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

Механіко-технологічний факультет

Кафедра кібербезпеки та програмного забезпечення

**ЛАБОРАТОРНА РОБОТА №7**

**Тема:** Контейнерні класи. Стандартна бібліотека шаблонів (STL) в С++

з дисципліни: Об’єктно-орієнтоване програмування

|  |
| --- |
| Виконав: ст. гр. КН-23 |
| Царенко С. О. |
| Перевірив: асистент  Козірова Н. Л. |
|  |

Кропивницький

2024

**Тема:** Контейнерні класи. Стандартна бібліотека шаблонів (STL) в С++

**Мета:** ознайомитись Контейнерні класи та навчитись їх програмно реалізовувати мовою С++.

**Варіант 5**

**Завдання**

Розробіть систему управління студентськими курсами. У цій системі потрібно зберігати та керувати інформацією про курси та студентів, які до них записалися. Використовуйте контейнерні класи зі стандартної бібліотеки шаблонів (STL) для збереження та операцій над даними. Використайте знання набуті в минулих заняттях для побудови програмного продукту.

• Додайте декілька курсів до списку, включаючи їх назву, викладача та кількість годин.

• Виведіть список курсів та кількості записаних студентів на кожний курс.

• Видаліть один або декілька курсів зі списку за назвою.

• Запишіть декілька студентів на певні курси.

• Виведіть оновлений список курсів та кількості записаних студентів на кожний курс.

• Виведіть список студентів, записаних на певний курс. Ви маєте розширити програму, додати додаткові методи та функціональні можливості, які вам здаються відповідними.

***Course.h:***

#ifndef COURSE\_H

#define COURSE\_H

#include <string>

#include <vector>

using namespace std;

class Course {

public:

Course(const string& courseName, const string& courseInstructor, int courseHours);

void addStudent(const string& studentName);

void removeStudent(const string& studentName);

int studentCount() const;

string getName() const;

string getInstructor() const;

int getHours() const;

const vector<string>& getStudents() const;

private:

string name;

string instructor;

int hours;

vector<string> students;

};

#endif // COURSE\_H

***Course.cpp:***

#include "Course.h"

#include <algorithm>

Course::Course(const string& name, const string& instructor, int hours) {

this->name = name;

this->instructor = instructor;

this->hours = hours;

}

void Course::addStudent(const string& studentName) {

students.push\_back(studentName);

}

void Course::removeStudent(const string& studentName) {

students.erase(remove(students.begin(), students.end(), studentName), students.end());

}

int Course::studentCount() const {

return students.size();

}

string Course::getName() const {

return name;

}

string Course::getInstructor() const {

return instructor;

}

int Course::getHours() const {

return hours;

}

const vector<string>& Course::getStudents() const {

return students;

}

***CourseManager.h:***

#ifndef COURSE\_MANAGER\_H

#define COURSE\_MANAGER\_H

#include "Course.h"

#include <vector>

using namespace std;

class CourseManager {

public:

void addCourse(const string& name, const string& instructor, int hours);

void removeCourse(const string& name);

void enrollStudent(const string& courseName, const string& studentName);

void listCourses() const;

void listStudentsInCourse(const string& courseName) const;

private:

vector<Course> courses;

Course\* findCourse(const string& name);

const Course\* findCourse(const string& name) const;

};

#endif // COURSE\_MANAGER\_H

***CourseManager.cpp:***

#include "CourseManager.h"

#include <iostream>

#include <iomanip>

void CourseManager::addCourse(const string& name, const string& instructor, int hours) {

courses.emplace\_back(name, instructor, hours);

}

void CourseManager::removeCourse(const string& name) {

courses.erase(

remove\_if(courses.begin(), courses.end(),

[&](const Course& course) { return course.getName() == name; }),

courses.end());

}

void CourseManager::enrollStudent(const string& courseName, const string& studentName) {

Course\* course = findCourse(courseName);

if (course) {

course->addStudent(studentName);

}

else {

cout << "Course \"" << courseName << "\" not found.\n";

}

}

void CourseManager::listCourses() const {

cout << "\nCourses List:\n";

cout << left << setw(20) << "Course Name"

<< setw(20) << "Instructor"

<< setw(10) << "Hours"

<< "Students Enrolled\n";

cout << string(60, '-') << "\n";

for (const auto& course : courses) {

cout << left << setw(20) << course.getName()

<< setw(20) << course.getInstructor()

<< setw(10) << course.getHours()

<< course.studentCount() << "\n";

}

}

void CourseManager::listStudentsInCourse(const string& courseName) const {

const Course\* course = findCourse(courseName);

if (course) {

cout << "\nStudents enrolled in \"" << course->getName() << "\":\n";

for (const auto& student : course->getStudents()) {

cout << "- " << student << "\n";

}

}

else {

cout << "Course \"" << courseName << "\" not found.\n";

}

}

Course\* CourseManager::findCourse(const string& name) {

for (auto& course : courses) {

if (course.getName() == name) {

return &course;

}

}

return nullptr;

}

const Course\* CourseManager::findCourse(const string& name) const {

for (const auto& course : courses) {

if (course.getName() == name) {

return &course;

}

}

return nullptr;

}

***Main.cpp:***

#include "CourseManager.h"

#include <iostream>

using namespace std;

int main() {

CourseManager manager;

manager.addCourse("Mathematics", "Dr. Smith", 30);

manager.addCourse("Physics", "Dr. Johnson", 40);

manager.addCourse("Computer Science", "Dr. Brown", 50);

manager.listCourses();

manager.enrollStudent("Mathematics", "Ben");

manager.enrollStudent("Mathematics", "Bob");

manager.enrollStudent("Physics", "Charlie");

manager.enrollStudent("Computer Science", "Roberto");

manager.listCourses();

manager.listStudentsInCourse("Mathematics");

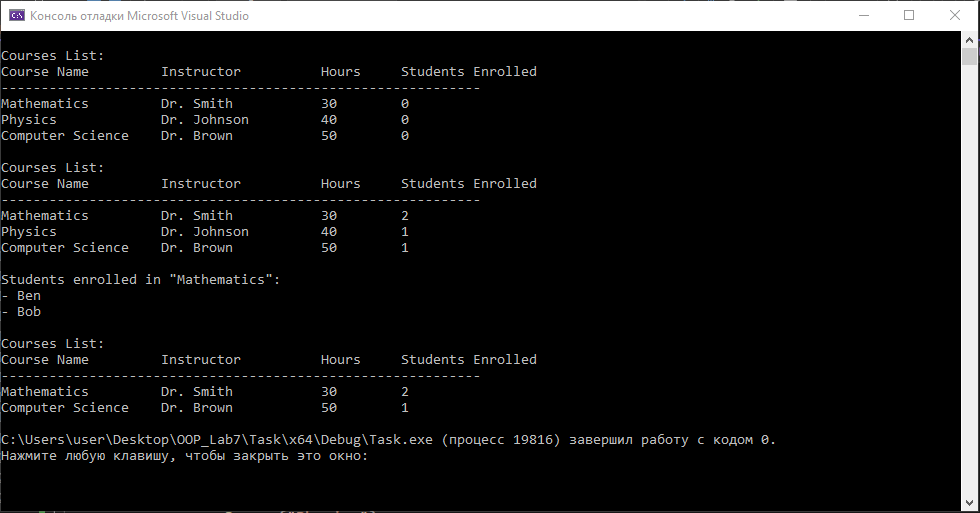
manager.removeCourse("Physics");

manager.listCourses();

return 0;

}

***Скріншот:***

****

**Висновок:**

У ході виконання лабораторної роботи №7 “ Контейнерні класи. Стандартна бібліотека шаблонів (STL) в С++”. Я ознайомився з поняттям Контейнерні класи та навчився їх програмно реалізовувати мовою С++.