**Lab 07**

**Object Oriented Programming Lab**

**Common Solution**

**Challenge-1:** *University Course Management System* --- (15 Marks)

**Header Files:**

**Student.h**

#ifndef STUDENT\_H

#define STUDENT\_H

#include"Course.h"

class Student

{

String name;

String ID;

Course\*\* courseList;

int coursesCount;

public:

Student(const String& name, int ID);

~Student();

String getName() const;

String getID() const;

void enrollCourse(Course& course);

void unEnrollCourse(const String& courseID);

void displayCourses() const;

};

#endif // !STUDENT\_H

**Course.h**

#ifndef COURSE\_H

#define COURSE\_H

#include"String.h"

#include<iostream>

using namespace std;

class Course

{

String name;

String ID;

public:

Course();

Course(const String& name, int ID);

String getName() const;

String getID() const;

void display() const;

};

#endif // !COURSE\_H

**OfferedCourses.h**

#ifndef OFFERED\_COURSES\_H

#define OFFERED\_COURSES\_H

#include"Course.h"

class OfferedCourses

{

Course courses[10];

int noOfCourses;

public:

OfferedCourses();

void displayOfferedCourses() const;

void addCourse(const Course& course);

const Course& getCourse(int index) const;

};

#endif // !OFFERED\_COURSES\_H

**.Cpp Files:**

**Student.cpp** ----- (7)

#include "Student.h"

Student::Student(const String& name, String ID) : name(name), ID(ID), courseList(nullptr), coursesCount(0) ----- (0.5)

{

}

Student::~Student()

{

delete[] courseList;

coursesCount = 0;

}

String Student::getName() const

{

return name;

}

String Student::getID() const

{

return ID;

}

void Student::enrollCourse(Course& course) ----- (1)

{

courseList[coursesCount] = &course;

coursesCount++;

}

Add pointer to the array. (0.5)

Increase the course count. (0.5)

void Student::unEnrollCourse(const String& courseID) ----- (4)

{

bool findFlag = false;

int i = 0;

while(i < coursesCount && !findFlag)

{

if (courseList[i]->getID() == courseID)

{

findFlag = true;

for (int j = i; j < coursesCount - 1; ++j)

{

courseList[j] = courseList[j + 1];

}

courseList[coursesCount - 1] = nullptr;

coursesCount--;

}

i++;

}

}

Search for course: 0.5

Remove course: 1

Move Course: 2

Keep course count: 0.5

Atomicity -1:

void Student::displayCourses() const ----- (1.5)

{

cout << "Student’s Enrolled Courses:\n";

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

{

cout << "Course " << i + 1 << ":\n";

courseList[i]->display();

}

}

Display in the right format: 0.75

Used display function of course for automicity: 0.75

**Course.cpp** ----- (3)

#include "Course.h"

Course::Course() :name(""), ID("") ----- (0.25)

{

}

Course::Course(const String& name, int ID) : name(name), ID(ID) ----- (0.25)

{

}

String Course::getName() const ----- (0.25)

{

return name;

}

String Course::getID() const ----- (0.25)

{

return ID;

}

void Course::display() const ----- (2)

{

cout << "Name: ";

name.display();

cout << "\nID: ";

ID.display();

cout << '\n';

}

Not Displayed in the right format: -1

**OfferedCourses.cpp** ----- (5)

#include "OfferedCourses.h"

OfferedCourses::OfferedCourses() : noOfCourses(0)

{

}

void OfferedCourses::displayOfferedCourses() const ----- (1.5)

{

cout << "Offered Courses:\n";

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

{

cout << "Course " << i + 1 << ":\n";

courses[i].display();

}

}

Display in the right format: 0.75

Used display function of course for automicity: 0.75

void OfferedCourses::addCourse(const Course& course) ----- (2)

{

if (noOfCourses < 10)

{

courses[noOfCourses++] = course;

}

}

Add pointer to the array. (0.5)

Check for noOfCourses. (0.5)

Increase the course count. (0.5)

const Course& OfferedCourses::getCourse(int index) const - (1.5)

{

if (!(index >= 0 && index < noOfCourses))

{

exit(0);

}

return courses[index];

}

Return course Index: 0.75

Check for valid index and exit. 0.75

**Quick Revision:**

**Student.cpp** ----- (7)

Student::Student(const String& name, String ID) ----- (0.25)

Student::~Student() ----- (0.25)

void Student::enrollCourse(Course& course) ----- (1)

Add pointer to the array. (0.5)

Increase the course count. (0.5)

void Student::unEnrollCourse(const String& courseID) ----- (4)

Search for course: 0.5

Remove course: 1

Move Course: 2

Keep course count: 0.5

Atomicity -1:

void Student::displayCourses() const ----- (1.5)

Display in the right format: 0.75

Used display function of course for automicity: 0.75

**Course.cpp** ----- (3)

Course::Course() :name(""), ID("") ----- (0.25)

Course::Course(const String& name, int ID) : name(name), ID(ID) ----- (0.25)

String Course::getName() const ----- (0.25)

String Course::getID() const ----- (0.25)

void Course::display() const ----- (2)

Not Displayed in the right format: -1

**OfferedCourses.cpp** ----- (5)

void OfferedCourses::displayOfferedCourses() const ----- (1.5)

Display in the right format: 0.75

Used display function of course for automicity: 0.75

void OfferedCourses::addCourse(const Course& course) ----- (2)

Add pointer to the array. (0.5)

Check for noOfCourses. (0.5)

Increase the course count. (0.5)

const Course& OfferedCourses::getCourse(int index) const - (1.5)

Return course Index: 0.75

Check for valid index and exit. 0.75

**Penalty Matrix:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Penalty List | Labs | | | | | | | | | | | | | | | |
| 1 | 2 | 3 | 3 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| Indentation putting { Infront of loop header, in do while, putting while with closing } | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |  |  |  |  |  |  |  |  |
| Meaningful Variable Names |  | -2 | -2 | -2 | -2 | -2 | -2 |  |  |  |  |  |  |  |  |  |
| Camel Case Notation | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |  |  |  |  |  |  |  |  |
| Atomicity |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Syntax error | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |  |  |  |  |  |  |  |  |
| Linker error | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |  |  |  |  |  |  |  |  |
| Wrong function prototypes | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |  |  |  |  |  |  |  |  |
| Class interface or additional members |  |  | 0 | 0 | 0 | 0 | 0 |  |  |  |  |  |  |  |  |  |
| Use of library function/class without permission | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |  |  |  |  |  |  |  |  |
| Continue statement | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |  |  |  |  |  |  |  |  |
| cin/cout where it isn’t needed | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |  |  |  |  |  |  |  |  |
| Multi-filing |  |  | 0 | 0 | 0 | 0 | 0 |  |  |  |  |  |  |  |  |  |
| Wrong #ifndef or name of header file |  |  | -2 | -2 | -2 | -2 | -2 |  |  |  |  |  |  |  |  |  |
| Global functions |  |  | -3 | -3 | -3 | -3 | -3 |  |  |  |  |  |  |  |  |  |
| Multiple classes in one header file |  |  | -3 | -3 | -3 | -3 | -50% |  |  |  |  |  |  |  |  |  |
| Function of preceding lab |  |  |  |  | -0.5 | -0.5 | -0.5 |  |  |  |  |  |  |  |  |  |