|  |  |
| --- | --- |
| **Total Marks:** | **4** |
| **Obtained Marks:** |  |

**Object Oriented Programming**

**Assignment # 01**

**Last date of Submission: October 12, 2023**

**Submitted To: Ms. Aniqa Shirazi**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Student Name: Kanwar Muhammad Jawad**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Reg Number: 2212227**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Attendence System:**

**Code:**

#include <iostream>

using namespace std;

const int MAX\_STUDENTS = 32;

class Student {

private:

std::string name;

int attendance;

public:

// Default constructor

Student() : name(""), attendance(0) {}

// Constructor with name

Student(string studentName) : name(studentName), attendance(0) {}

void markAttendance() {

attendance++;

}

int getAttendance() const {

return attendance;

}

string getName() const {

return name;

}

};

class AttendanceManagement {

private:

Student students[MAX\_STUDENTS];

public:

// Default constructor

AttendanceManagement() {

for (int i = 0; i < MAX\_STUDENTS; ++i) {

students[i] = Student(); // Initialize with default constructor

}

}

void addStudent(string name, int index) {

if (index >= 0 && index < MAX\_STUDENTS) {

students[index] = Student(name);

} else {

cout << "Invalid index for adding student." << endl;

}

}

void markAttendance(int studentIndex) {

if (studentIndex >= 0 && studentIndex < MAX\_STUDENTS) {

students[studentIndex].markAttendance();

} else {

cout << "Invalid student index." << endl;

}

}

void displayAttendance(int studentIndex) const {

if (studentIndex >= 0 && studentIndex < MAX\_STUDENTS) {

cout << "Attendance for " << students[studentIndex].getName() << ": "

<< students[studentIndex].getAttendance() << endl;

} else {

cout << "Invalid student index." << endl;

}

}

};

int main() {

AttendanceManagement attendanceManager;

char choice;

int studentIndex;

std::string studentName;

do {

cout << "Menu:\n";

cout << "1. Add a student\n";

cout << "2. Mark attendance\n";

cout << "3. Display attendance\n";

cout << "4. Exit\n";

cout << "Enter your choice: ";

cin >> choice;

switch (choice) {

case '1':

cout << "Enter student name: ";

cin >> studentName;

cout << "Enter student index (0-31): ";

cin >> studentIndex;

attendanceManager.addStudent(studentName, studentIndex);

break;

case '2':

cout << "Enter student index (0-31): ";

cin >> studentIndex;

attendanceManager.markAttendance(studentIndex);

break;

case '3':

cout << "Enter student index (0-31): ";

cin >> studentIndex;

attendanceManager.displayAttendance(studentIndex);

break;

case '4':

cout << "Exiting program.\n";

break;

default:

cout << "Invalid choice. Try again.\n";

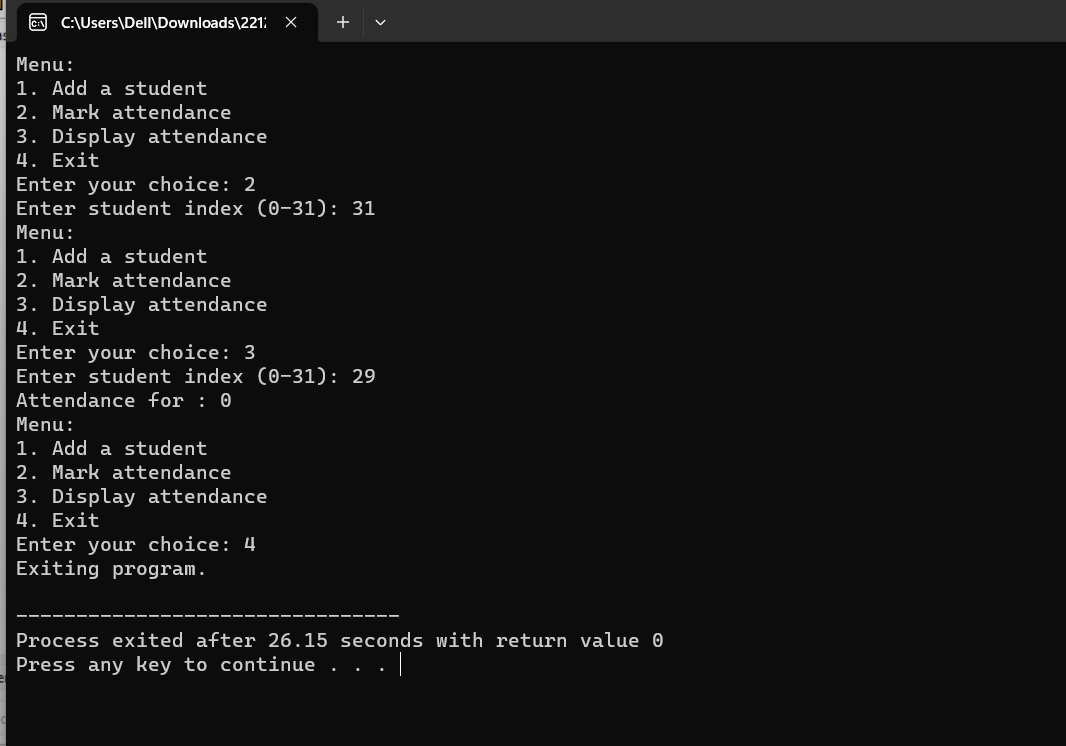
}

} while (choice != '4');

return 0;

}

**Output:**

****