**SALEM MAINA**

**SCT212-0167/2022**

**DATA STRUCTURES AND ALGORITHMS LAB 1 ASSIGNMENT**

1. **Functions:**

* **summation**: Calculates the sum of integers in an array.
* **maximum**: Finds the largest integer in the array.

**Steps:**

* Declare two functions **summation** and **maximum**.
* Inside the **main** function:

a. Declare an array of integers of length **n**.

b. Prompt the user to input the value of **n**.

c. Allow the user to input **n** integers into the array.

d. Call the **summation** function passing the array and display the sum.

e. Call the **maximum** function passing the array and display the maximum value.

C program

#include <iostream>

#include <limits>

int summation(int arr[], int n) {

int sum = 0;

for (int i = 0; i < n; ++i) {

sum += arr[i];

}

return sum;

}

int maximum(int arr[], int n) {

int max = arr[0];

for (int i = 1; i < n; ++i) {

if (arr[i] > max) {

max = arr[i];

}

}

return max;

}

int main() {

int n;

std::cout << "Enter the length of the array: ";

std::cin >> n;

if (n <= 0 || n > std::numeric\_limits<int>::max()) {

std::cout << "Invalid array length\n";

return 1;

}

int arr[n];

std::cout << "Enter " << n << " integers:\n";

for (int i = 0; i < n; ++i) {

std::cin >> arr[i];

}

std::cout << "Sum of the integers: " << summation(arr, n) << std::endl;

std::cout << "Maximum integer: " << maximum(arr, n) << std::endl;

return 0;

}

**2**

**Structs:**

**Student** struct containing:

* + Registration number (string)
  + Name (string)
  + Age (integer)
  + Course (Course struct)
  + Grades (Grade struct)

**Course** struct containing:

* + Course code (string)
  + Course name (string)

**Grade** struct containing:

* + Mark (integer)
  + Grade (char)

**Operations:**

1. Add a student.
2. Edit student details.
3. Add marks and calculate grades (using the grading system).
4. Ensure grades cannot be altered after calculation.

**C++ Classes:**

**Student** class with private member variables for registration number, name, age, course, and grades, and public member functions for operations like adding a student, editing details, adding marks, and calculating grades.

**Course** class with private member variables for course code and course name.

**Grade** class with private member variables for mark and grade.

**Steps:**

1. Define classes **Student**, **Course**, and **Grade**.
2. Implement member functions for each class to perform operations described.

#include <iostream>

#include <string>

#include <vector>

using namespace std;

// Grade structure

struct Grade {

int mark;

char the\_grade;

};

// Course structure

struct Course {

string course\_code;

string course\_name;

};

// Student structure

struct Student {

string registration\_number;

string name;

int age;

Course course;

Grade grades;

bool grades\_calculated = false; // Flag to indicate if grades are calculated

};

// Function to calculate grade based on mark

char calculateGrade(int mark) {

if (mark > 69)

return 'A';

else if (mark > 59)

return 'B';

else if (mark > 49)

return 'C';

else if (mark > 39)

return 'D';

else

return 'E';

}

// Function to add a new student

void addStudent(vector<Student>& students) {

if (students.size() >= 40) {

cout << "Maximum number of students reached (40)." << endl;

return;

}

Student new\_student;

cout << "Enter registration number: ";

cin >> new\_student.registration\_number;

cout << "Enter name: ";

cin >> new\_student.name;

cout << "Enter age: ";

cin >> new\_student.age;

cout << "Enter course code: ";

cin >> new\_student.course.course\_code;

cout << "Enter course name: ";

cin >> new\_student.course.course\_name;

students.push\_back(new\_student);

cout << "Student added successfully." << endl;

}

// Function to edit student's details

void editStudent(vector<Student>& students) {

string reg\_number;

cout << "Enter registration number of the student to edit: ";

cin >> reg\_number;

for (auto& student : students) {

if (student.registration\_number == reg\_number) {

cout << "Enter new name: ";

cin >> student.name;

cout << "Enter new age: ";

cin >> student.age;

cout << "Enter new course code: ";

cin >> student.course.course\_code;

cout << "Enter new course name: ";

cin >> student.course.course\_name;

cout << "Student details updated successfully." << endl;

return;

}

}

cout << "Student with registration number " << reg\_number << " not found." << endl;

}

// Function to add marks and calculate grades

void addMarks(vector<Student>& students) {

string reg\_number;

cout << "Enter registration number of the student: ";

cin >> reg\_number;

for (auto& student : students) {

if (student.registration\_number == reg\_number) {

cout << "Enter mark for " << student.name << ": ";

cin >> student.grades.mark;

student.grades.the\_grade = calculateGrade(student.grades.mark);

student.grades\_calculated = true;

cout << "Marks and grades added successfully." << endl;

return;

}

}

cout << "Student with registration number " << reg\_number << " not found." << endl;

}

int main() {

vector<Student> students;

// Menu-driven program

int choice;

do {

cout << "\nStudent Management System" << endl;

cout << "1. Add Student" << endl;

cout << "2. Edit Student Details" << endl;

cout << "3. Add Marks and Calculate Grades" << endl;

cout << "4. Exit" << endl;

cout << "Enter your choice: ";

cin >> choice;

switch (choice) {

case 1:

addStudent(students);

break;

case 2:

editStudent(students);

break;

case 3:

addMarks(students);

break;

case 4:

cout << "Exiting program." << endl;

break;

default:

cout << "Invalid choice. Please enter a valid option." << endl;

}

} while (choice != 4);

return 0;

}