



CPP-A11: Struct, Union, and Enum

Assignment: Console-Based Student Registration System

Problem Statement

Design a **Student Registration System** where the system allows users to:

- Add new students.
- View all registered students.
- Search for a student by their ID.
- Update the student details.
- Delete a student from the system.

Each student will have the following attributes:

- **Student ID** (unique identifier)
- **Name**
- **Age**
- **Department**
- **User Type** (Enum: **STUDENT**, **ADMIN**, **TEACHER**)
- **Additional Info** (Union: Scholarship Amount for students, Salary for teachers, and System Permissions for admins)

The system should store all student records in a vector and provide options for different operations via a menu-driven interface.

Requirements

1. Struct **Student** Definition:

- **studentId** : integer

- `name` : string
- `age` : integer
- `department` : string
- `userType` : enum (UserType: `STUDENT` , `ADMIN` , `TEACHER`)
- `additionalInfo` : union
- `scholarshipAmount` (for `STUDENT` userType)
- `salary` (for `TEACHER` userType)
- `permissions` (for `ADMIN` userType)

2. Enum Definition:

```
enum UserType { STUDENT, ADMIN, TEACHER };
```

3. Union Definition:

```
union AdditionalInfo {
    double scholarshipAmount;
    double salary;
    string permissions;
};
```

4. Features to Implement:

- **Add Student:** Create a function to accept student details and push them into the vector.
- **View Students:** Display a list of all students with their details.
- **Search Student:** Find a student by their `studentId` and display their details.
- **Update Student:** Allow updating any attribute of a specific student.
- **Delete Student:** Remove a student record from the vector.
- Use meaningful function names and modular code.

Expected Input/Output

Menu Options:

1. Add Student
 2. View All Students
 3. Search Student by ID
 4. Update Student by ID
 5. Delete Student by ID
 6. Exit
- Enter your choice: _

Example Interaction:

```
1. Add Student
Enter Student ID: 101
Enter Name: John Doe
Enter Age: 20
Enter Department: Computer Science
Enter User Type (0: STUDENT, 1: ADMIN, 2: TEACHER): 0
Enter Scholarship Amount: 2000

2. View All Students
ID: 101, Name: John Doe, Age: 20, Department: Computer Science, UserType: STUDENT, Scholarship Amount: 2000

3. Search Student by ID
Enter Student ID to Search: 101
ID: 101, Name: John Doe, Age: 20, Department: Computer Science, UserType: STUDENT, Scholarship Amount: 2000

4. Update Student by ID
Enter Student ID to Update: 101
Enter New Age: 21

5. Delete Student by ID
Enter Student ID to Delete: 101
Student Deleted Successfully
```

Happy Coding!