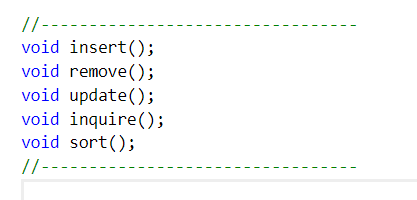
**Problem definition and condition:**

**Description**: Design and implement a system to manage the information of students. The system contains the ID, name, age, gender information for each student; the course information contains the course ID, course name, marks for each student, and professor. The system should have the function of **insert, remove, update, inquire, sort** and count all the information.

**Requirements**:

* The system should record information listed in the above description.
* The information can be store in the file or only exists in the prime memory.
* They system should be able to insert, remove, update, inquire, sort and count both the student information and the course information.
* Sorting by the marks for one course should be implemented.

**There few function:**

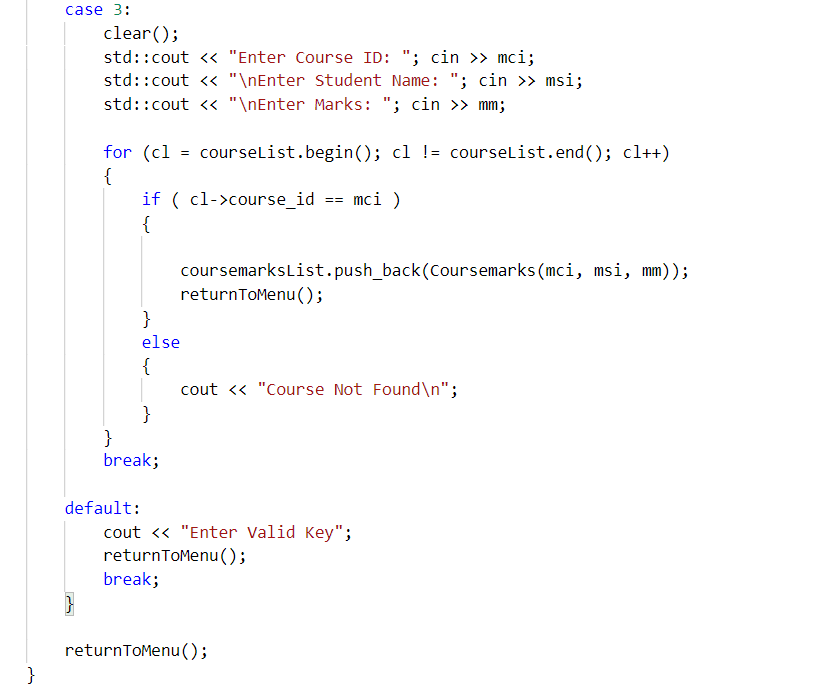
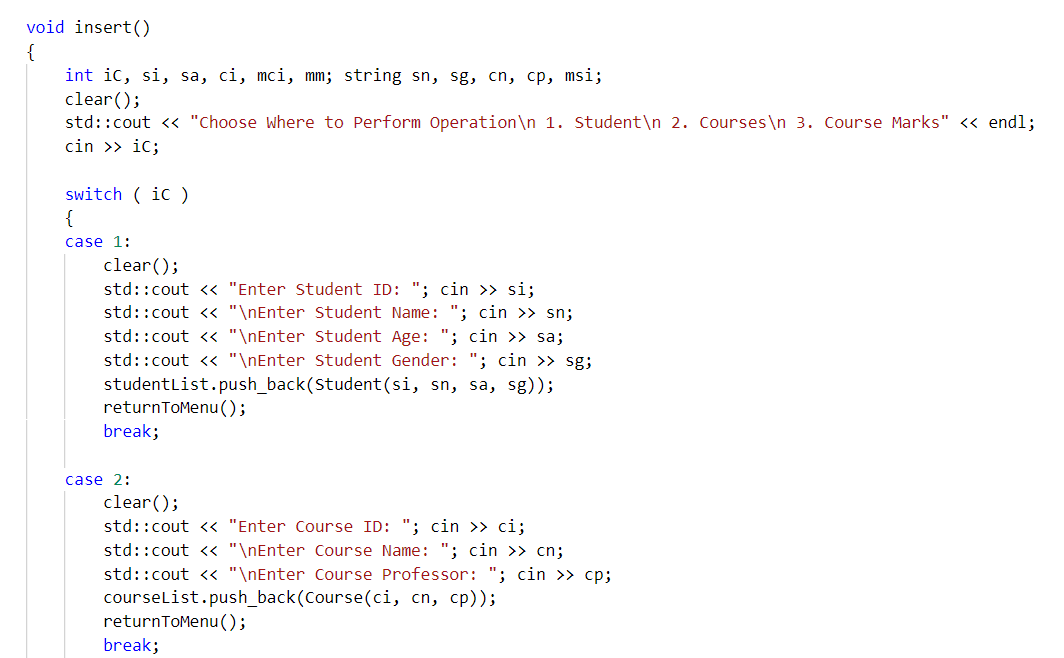


* Insert
* Remove
* Update Info
* Inquire
* Sort Lists
* Exit

**Each section has few function to operate:**

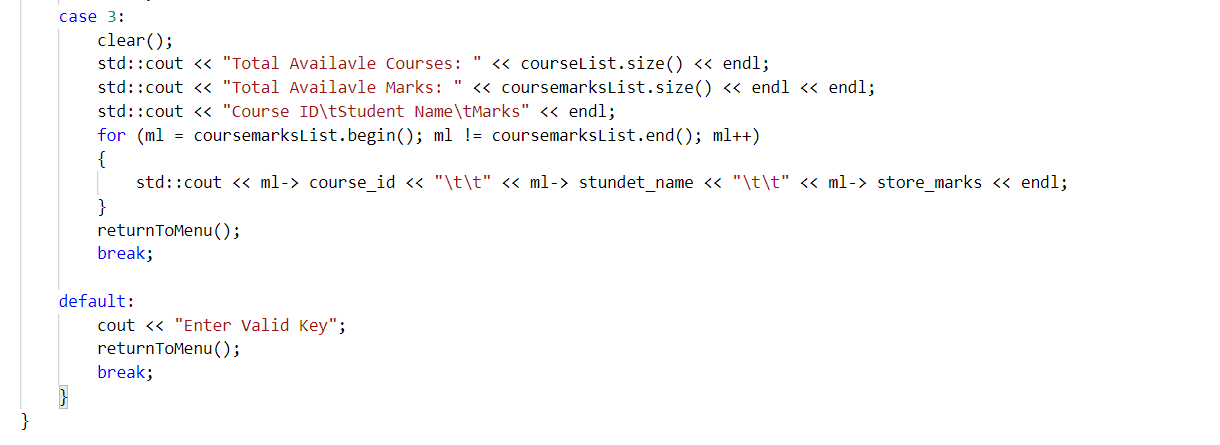
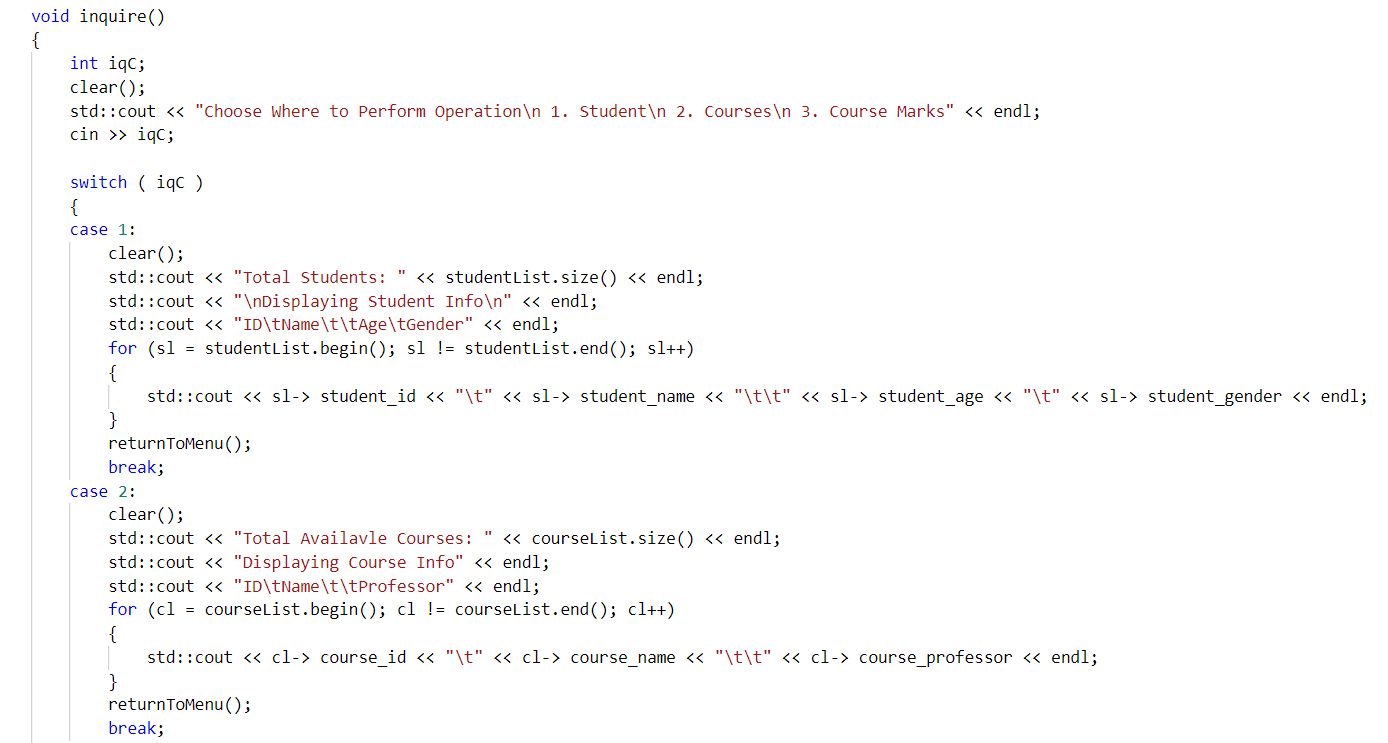
**1. Insert:**

* Student
* Course
* Course Marks



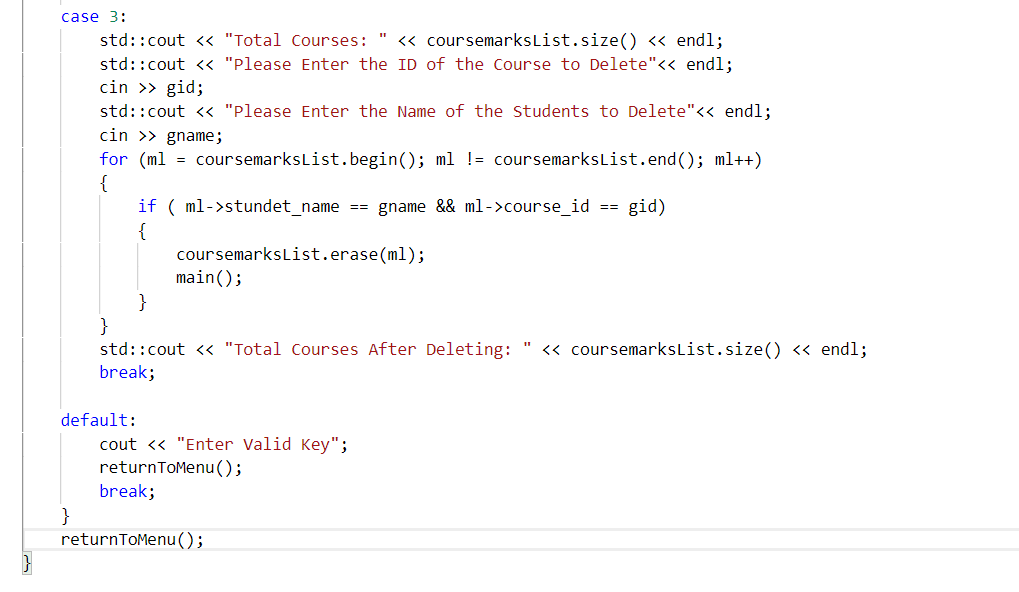
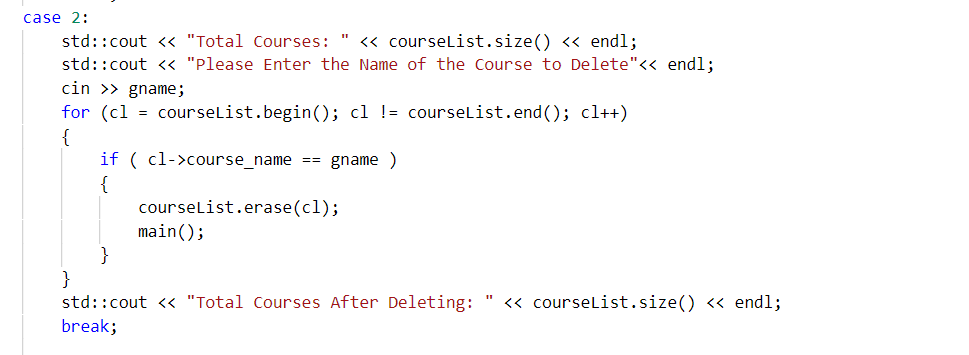
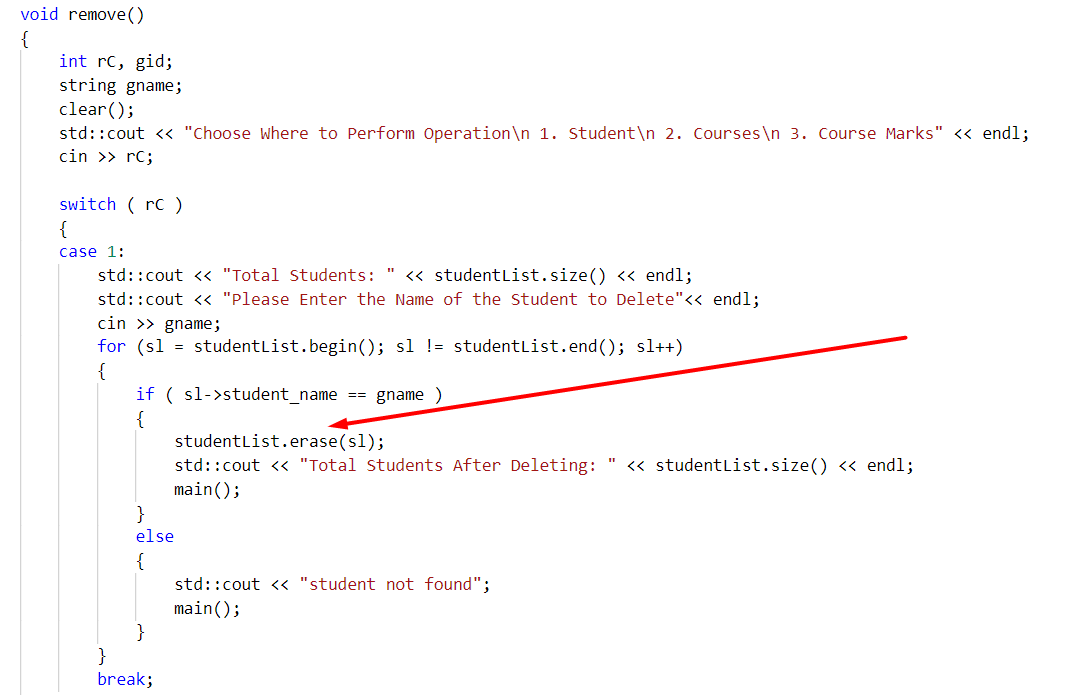
**2. inquire:**

* Student
* Course
* Course Marks



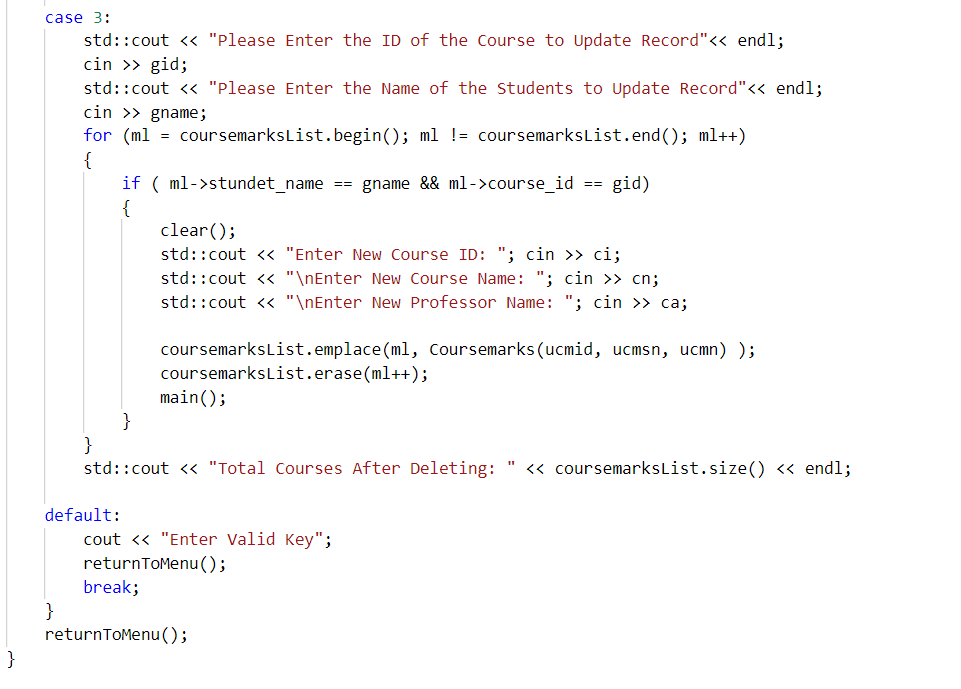
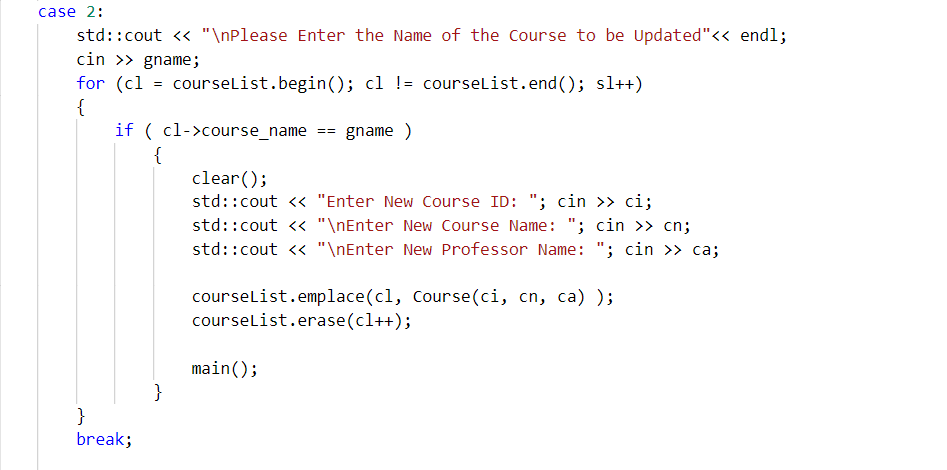
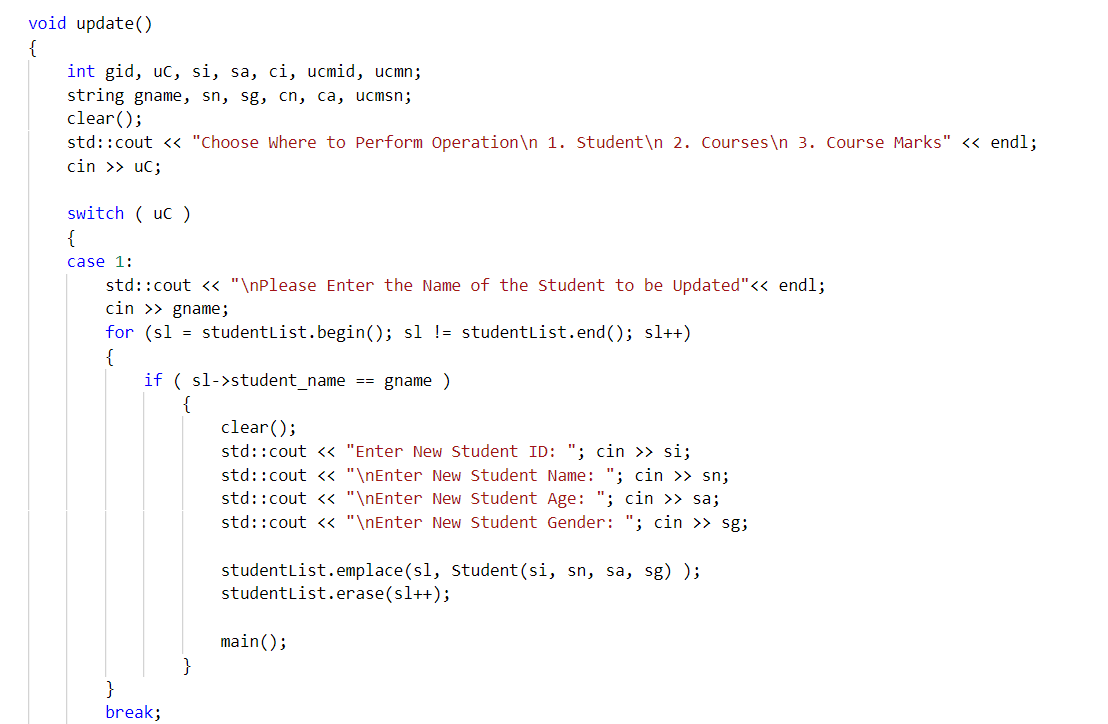
**3. remove info:**

* Student
* Course
* Course Marks



**3. Update info:**

* Student
* Course
* Course Marks



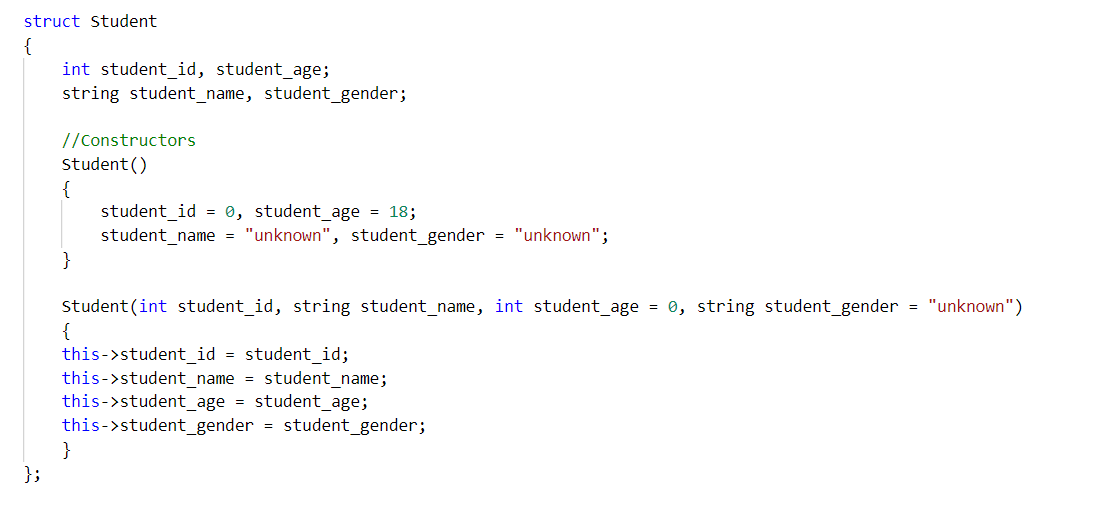
**4. Sort Lists:**

* Sort student by Name
* Sort Course by ID
* Sort Course Marks by Marks

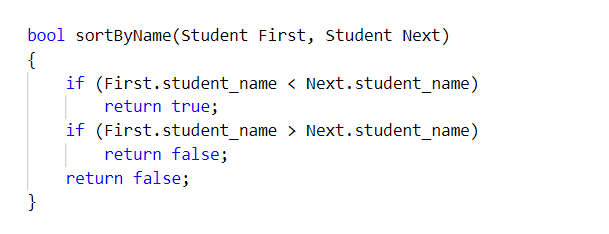


5. Exit:

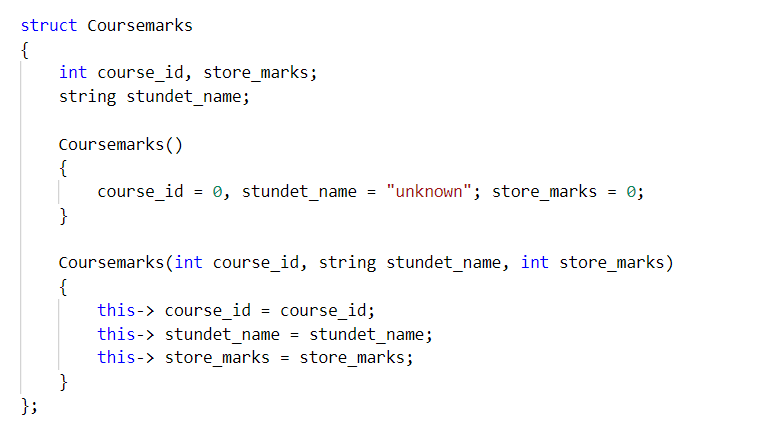
**Student class:**



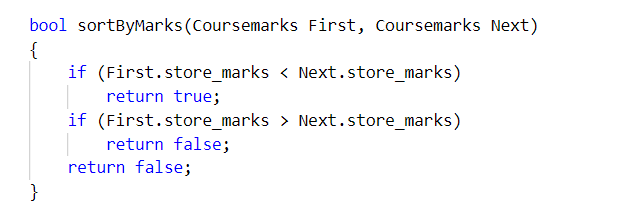
**Sort algorithm: Binary search Tree (**log(n)**)**



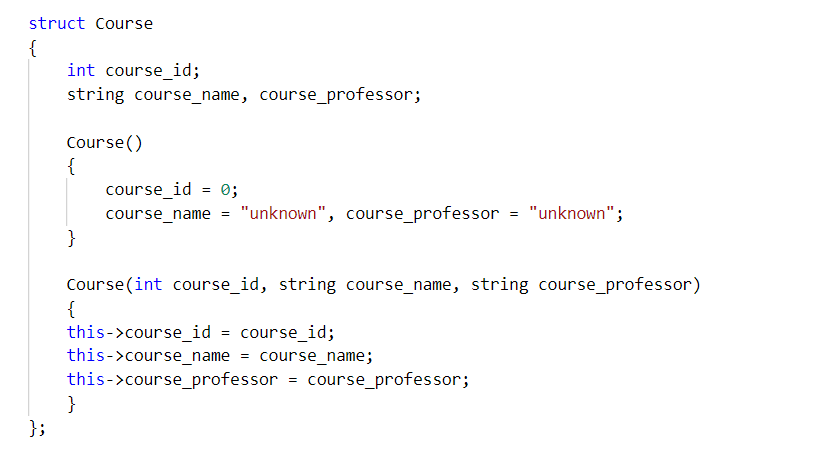
**Course marks:**



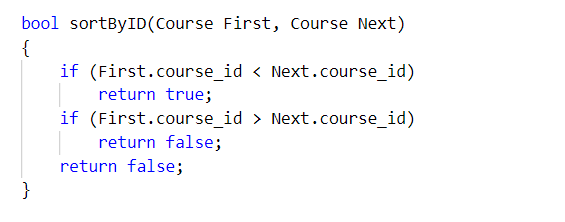
**Sort algorithm: Binary search Tree (**log(n)**)**



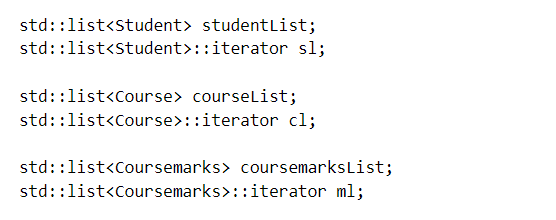
**Course:**



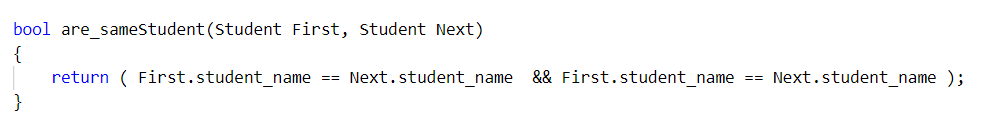
**Sort algorithm: Binary search Tree (**log(n)**)**



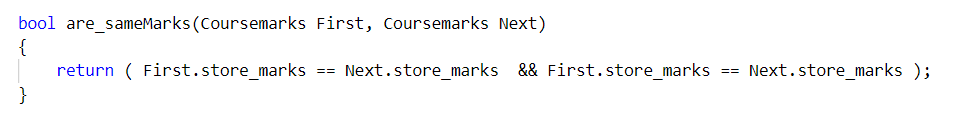
**Student, course, course-marks insert on List:**



**If Student has same name**



**If Course-Marks has same marks**



**If course has same ID**

