An-Najah National University
Faculty of Engineering and Information Tech.
Electrical and Computer Engineering Department
Data Structures and Algorithms (10636211)
HW 1

Due to: 18/10/2022 **20 points**

Write C++ program to represent the following classes: Student, Course, and Registrar.
Each class has its own data and functions as specified below.

Course class:

private information:

- course name
- Vector<Student*> students [here you need to practice the concept of template by using Vector in C++ STL (standard template library)
  https://www.geeksforgeeks.org/vector-in-cpp-stl/

public information:

1. A user defined constructor that takes the name of course as a parameter.
2. A destructor to clear the memory.
3. getName(); function that returns the name of course.
4. addStudent(Student* s); that adds any given Student to the vector of students.
5. removeStudentsFromCourse(); that removes all students from the course.
6. Overload the insertion operator <<. This operator should print the contents of the object.

Student class:

private information:

- student name
- vector<Course*> courses

public information:

1. A user defined constructor that takes the name of the student as a parameter.
2. A destructor to clear the memory.
3. getName(); function that returns the name of the student.
4. A boolean function addCourse(Course* C); that adds any given course to the vector of courses
5. A void function removedFromCourse(Course* C); that removes a student from a given course.
6. Overload the insertion operator <<. This operator should print the contents of the object.

Registrar class:

private information:

1. A vector<Course*> courses;
1. A vector<Student*> students;
2. An integer function findStudent(studentName); that returns the index of student if found in the vector students , else return -1
3. An integer function findCourse(courseName) ; that return the index of course if found in the vector courses, else return -1

public information:

1. A default constructor Registrar();
2. A boolean function addCourse(courseName); that creates a new course and adds it to the courses vector.
3. A boolean function addStudent(studentName); ; that creates a new student and adds it to the students vector.
4. A boolean function enrollStudentInCourse(studentName,courseName); this function should check if both course and student are found and then assign the course to that student, else the operation fails.
5. A boolean function cancelCourse(courseName); this function should check if the course is found and then remove all students from that course.
6. A void function clear(); that clears both courses and students vectors to prepare for the next semester.

Main should test the following code:

```
int main(){

        Registrar registrar;

         cout << "No courses or students added yet\n";
         cout << registrar << endl;

         cout << "AddCourse DS10636211\n";
         registrar.addCourse("DS10636211");
         cout << registrar << endl;

         cout << "AddStudent Ali_Ahmad\n";
         registrar.addStudent("Ali_Ahmad");
         cout << registrar << endl;
```

```
        cout << "AddCourse Java10636212\n";
        registrar.addCourse("Java10636212");
        cout << registrar << endl;

        cout << "EnrollStudentInCourse Ali_Ahmad DS10636211\n";
        registrar.enrollStudentInCourse("Ali_Ahmad", "DS10636211");
        cout << "EnrollStudentInCourse Ali_Ahmad Java10636212\n";
        registrar.enrollStudentInCourse("Ali_Ahmad", "Java10636212");
        cout << registrar << endl;

        cout << "EnrollStudentInCourse Haya_Samaana DS10636211\n";
        cout << "Should fail since Haya_Samaana is not a student.\n";
        registrar.enrollStudentInCourse("Haya_Samaana", "DS10636211");
        cout << registrar << endl;

        cout << "CancelCourse DS10636211\n";
        registrar.cancelCourse("DS10636211");
        cout << registrar << endl;

        cout << "Clear system to start  new semester\n";
        registrar.clear();
        cout << registrar << endl;

    return 0;

}
```