

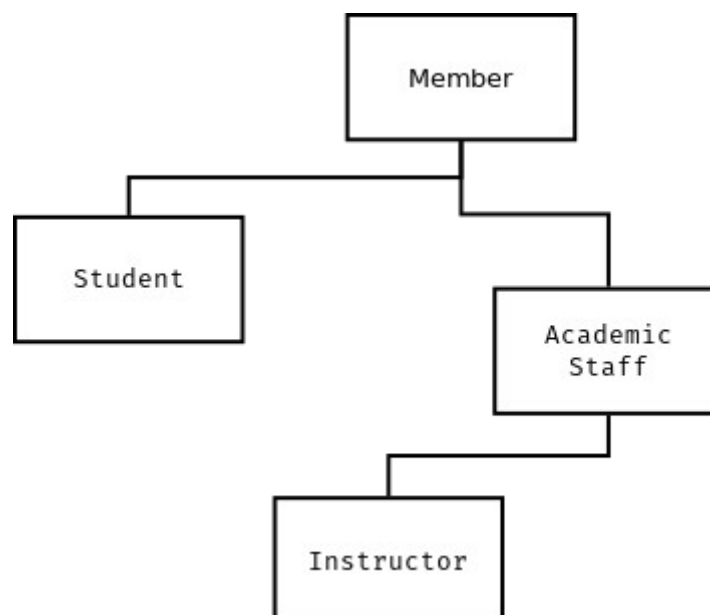
**Lab Assignment 5 (2 Week)****2020 – 2021 Fall, CMPE 211 Fundamentals of Programming II****A Better Course Registration System**

For this assignment, you are asked to design and implement a course registration system that handles bookkeeping of both academic staff and students using object oriented design.

The classes are as follows;

- **Course**
- **Member**
- **Student (will not be used this week)**
- **AcademicStaff**
- **Instructor**

and a **Driver** class to wrap everything together. You are asked to make use of inheritance between your classes. Here's a handy diagram;



The requirements of the system are:

- Each **Course** has at most one instructor. You need to check whether the **Course** is assigned to somebody else at any time.
- Each **Course** has a year.
- Each **Course** has a semester; Fall, Spring, or Summer.
- Each **Academic Staff** has a list of **Courses** (of your **Course** class)
- Each **Academic Staff** has an office number.
- **Instructors** are not allowed to teach more than 3 **Courses**.
- **Instructors** have academic titles; Lecturer, Assistant Professor, Associate Professor, Professor.
- Optional: You can use enumerated types for semester and academic titles.

Your driver class needs to get inputs from user to perform following operations:

- Create a new **Course**

```
> create;Course;<Course Code>;<Course Name>;<Year>;<Semester>;<Course Capacity>
```

- Create a new **Instructor**

```
> create;Instructor;<Instructor Name>;<Instructor ID>;<Office Number>;<Title>
```

- Assign an **Instructor** to a **Course**

```
> assign;Instructor;<Instructor ID>;<Course Code>
```

- Print the information about an **Instructor** with all the **Courses** they are assigned to.

```
> report;Instructor;<Instructor ID>
```

- Print the **Course** information with **Instructor** names, offices etc.

```
> report;Course;<Course Code>
```

The separator between the fields is “;” because instructor names, course names etc. might have spaces (“ ”) between them.

Your application should do the following checks:

- If the **Course** or **Instructor** already exists while the user is attempting to create a new one, the program should print “Instructor/Course already exists” error message (for the matching case).
- If the **Course** has already assigned an **Instructor**, the program should print “<Instructor ID> - <Name Surname> is already teaching <Course Code> – <Course Name>” as an error message.
- The number of **Courses** assigned to an **Instructor** per semester should not exceed 3. If user tries to add a 4<sup>th</sup> course, “Instructor <Instructor ID> - <Name Surname> cannot teach more than 3 courses in this semester” should be printed as an error message.
- If the operations are done successfully, the program should print “success” in a separate line. In case of any error, it should print the appropriate error message. Typos on error messages can be handled with “Unknown command”.