## East West University

**Department of Computer Science and Engineering**

## A/2, Jahurul Islam Avenue, Jahurul Islam City, Aftabnagar, Dhaka

**Lab Manual:** 05

**Lab Topic:** Class Relationships

**Course Code:** CSE110 (Object Oriented Programming)

**Course Instructor:** Tanni Mittra, Senior Lecturer, CSE

**Lab Objective**

1. **Familiarize** students with the implementation of classes
2. **Write** various instance methods performing different actions on the objects of a class
3. **Write** the definition of multiple classes and
4. Can **identify** and **hold** their relationships among each other.

**Lab Activities**

1. **Defining Multiple Classes** 
   * + We want to develop a minimal, simple object-oriented application for a university.
     + A university has three major entities: Students, Faculties and Courses.
     + First, we have to identify the relationships among them.
     + The following relationship diagram shows the relationships among Student, Course and Faculty class.

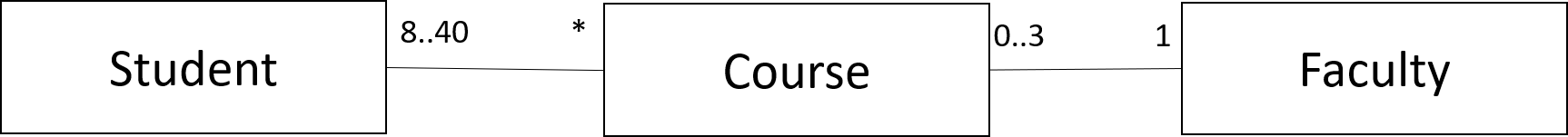


Figure1: Relationship among Student, Course and Faculty

**Problem\_1:** Your job is to define the above-mentioned classes as per the specification mentioned below and then write a Main/Driver class that demonstrates the functionalities of these classes.

|  |  |  |
| --- | --- | --- |
| **Student** | **Course** | **Faculty** |
| - studentId: int  -studentName: String  -studentCGPA: double | - courseId: String  - courseTitle: String  - credit: double  - studentList: Student []  - numberOfStudents: int  - faculty: Faculty | - facultyId: int  - facultyName: String  - facultyPosition: String |
| + Student()  + Student(studentId, studentName, studentCGPA)  + toString(): String | + Course()  + Course(courseId, courseTitle, credit)  + toString(): String  + addStudent(Student): void  + dropStudent(studentId): void  + addFaculty(Faculty): void  + dropFaculty(): void  + printStudentList(): void | + Faculty()  +Faculty(facultyId, facultyName, facultyPosition)  + toString(): String |

**Assignment:**  **Developing a Menu-based Application**

* Now, we need to develop a menu-based application.
* The initial menu may have the following options:
  1. Add
  2. Delete
  3. Update
  4. Search
* For each of these options, we may provide further options. Suppose, for ‘Add’ option, next we may show the following options:
  1. Add a Student
  2. Add a Course
  3. Add a Faculty

For ‘Delete’ and ‘Update’, we may provide the same options.

* Search is very important feature in our application. For ‘Search’ option, we may provide the followings:
  1. Search a Student
  2. Search a Course
  3. Search a Faculty
  4. Search whether a student takes a course
  5. Search whether a faculty teaches a course
  6. Search courses taken by a student
  7. Search courses taught by a faculty