Class Relationships

Lab Objective

Familiarize students with the implementation of multiple classes and their relationships among each other.

Lab Outcomes

After completing this lab successfully, students will be able to:

1. **Write** the definition of multiple classes and can **identify** and **hold** their relationships among each other.
2. **Use** NetBeans IDE for developing moderately complex object-oriented applications.

Psychomotor Learning Levels

This lab involves activities that encompass the following learning levels in psychomotor domain.

Level	Category	Meaning	Keywords
P1	Imitation	Copy action of another; observe and replicate.	Relate, Repeat, Choose, Copy, Follow, Show, Identify, Isolate.
P2	Manipulation	Reproduce activity from instruction or memory	Copy, response, trace, Show, Start, Perform, Execute, Recreate.

Lab Activities

A. 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.

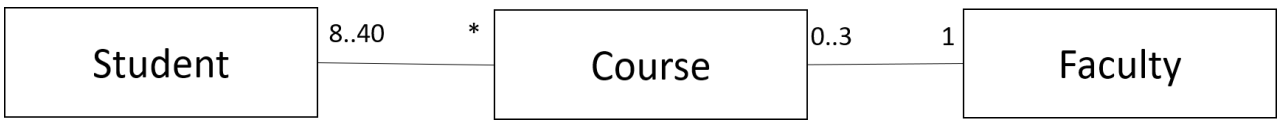

Figure 1: Relationship among Student, Course and Faculty

- **<u>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.</u>**

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(studentId): void + dropStudent(studentId): void + addFaculty(facultyId): void + dropFaculty(): void + printStudentList(): void	+ Faculty() + Faculty(facultyId, facultyName, facultyPosition) + toString(): String

B. Developing a Menu-based Application

- Now, we need to develop a menu-based, command line application.
- The initial menu may have the following options:
    a. Add
    b. Delete
    c. Update
    d. Print
    e. Search
- For each of these options, we may provide further options. Suppose, for 'Add' option, next we may show the following options:
    a. Add a Student
    b. Add a Course
    c. Add a Faculty

  For 'Delete' and 'Update', we may provide the same options.
- For 'Print' option, we may further provide the followings:
    a. Print all students
    b. Print all course
    c. Print all faculties
    d. Print information of a student
    e. Print information of a course
    f. Print information of a faculty
    g. Print student list and faculty information of a course
    h. Print courses taken by a student
- Search is very important feature in our application. For 'Search' option, we may provide the followings:
    a. Search a Student
    b. Search a Course
    c. Search a Faculty
    d. Search whether a student takes a course
    e. Search whether a faculty teaches a course
    f. Search courses taken by a student
    g. Search courses taught by a faculty