

# MET CS665 Final Project

© 2023, Suresh Kalathur, Boston University, All Rights Reserved.

The following document should not be disseminated outside the purview of its intended purpose.

The Computer Science department offers the following programs:

- Degree Programs -- Undergraduate: 4 year Bachelor of Science (B.S.) in CS (Computer Science) and CIS (Computer & Information Systems); Graduate: 2 year Master of Science (M.S.) in CS, CIS, and DA. Each of these programs requires a fixed number of core courses (determined by the department) and fixed number of elective courses (the student picks the electives from the choices particular to the program). Let us assume that a year is made up of two semesters. For the B.S. and the M.S. programs, students take only the electives in their last year of study. For both the programs, students do an additional thesis in their last semester.
- Certificate Programs in various concentrations (Security, Web Technology, Analytics, etc) -- each program requires 4 courses. These programs have no time limit and a student gets the certificate whenever the course requirements are met.

The department has a Chairperson, full time faculty, and part time faculty. The Chairperson and the part time faculty teach only one course per semester, where as the full time faculty teach three courses each semester. Among the full time faculty, there will be a single Graduate advisor and a single Undergraduate Advisor. The advisors serve in that capacity for a year. Students can send a query to their advisor as well as the Chairperson and any faculty. Full time faculty also serve as the thesis advisor for the students.

The courses offered by the department are grouped under various concentration areas (e.g., Systems, Programming Languages, Databases, Software Engineering, etc.). A concentration would have a collection of courses, or if the concentration is a wide one, it will be organized into a hierarchical structure with sub-concentrations. For example, the Programming Languages concentration could have sub-concentrations like Procedural Languages, Object Oriented Languages, Functional Languages, and so on. Let us assume that a given course is taught by the same faculty person each and every time. A full-time faculty member is responsible to coordinate each concentration (or sub-concentration). A full-time faculty member can be the coordinator for more than one concentration (or sub-concentration). The overall responsibility for the top-level concentrations lies with the chairman.

Each course has a description, syllabus, and a method named *format* which returns an HTML format output of the course. Each concentration also has the description and the method *format* which recursively calls the format of its components and returns an HTML format output for the concentration.

For a given student, we should be able to tell what program the student is enrolled in, what courses are taken by the student each semester, what their thesis is and the thesis advisor (if

applicable). Also, provide the method, *gpa*, which computes the GPA of the student (assume each course is 4 credits and the following points for the grades: A (4), B (3), C (2), D (1), F(0))

The Department keeps track of the courses offered each semester. The department also wants to know the students enrolled in a given course for a particular semester.

A faculty member should easily know what courses they are teaching a given semester, and the students they are advising for that semester.

Students enroll for courses for a given semester with the department. Each course has an enrollment limit and when the limit is reached, the Chairperson will be notified automatically. Students will be wait-listed when the course limit is reached. When a student drops from such a course, the first person in the wait list for that course will be notified and enrolled automatically.

- **Show the class diagram without considering any design patterns. Show the associations between the classes with navigability and multiplicity. Show the attributes and operations for the classes.**
- **Implement the Java solution for the above making use of at least two creational patterns, at least two structural patterns, and at least two behavioral patterns. Show the class diagrams for the design patterns individually.**
- **Demonstrate students enrolling/dropping in classes, and the notification to the chairperson when an enrollment limit is reached. Also demonstrate the case when student is wait-listed and an opening occurs for that course. You can hard code the data about the semesters, courses, and enrollment limits without having a need for an explicit database.**

### **Submission:**

- **Submit one Zip file of the IntelliJ IDEA project or Eclipse project**
- **The zip file should be named cs665\_finalproject\_lastName.zip**
- **Upload the zip file to the FinalProject submission**