Instructions: The homework is due on Fri, Apr 22, 23:59:59. Submission is done in electronic format, create a folder HW4 in your course directory and place source code and compilation scripts there. Have a separate folder for each question (Q1 and Q2). Recall to set the proper permissions for files! (files must be owned by grader group, and must not be readable by others). Comments in all files are required, in order to make the code readable. Also, ensure that you treat error/exception cases properly in your programs.

Requirements: You must create an application for managing the courses offered by a university. The schema is as follows:

```sql
Faculty(fid, fname, dept)
Students(sid, sname)
Courses(cid, cname, credits, fid)
Enrolled(sid, cid, grade)
```

The Faculty relation stores data about faculty members, and has a faculty unique identifier fid, the faculty name and a department (string). Similarly, students have a student id and name. Each course has a course id, name, credits, and the id of the faculty that teaches the course (you must reflect this properly in the table definition). Assume that there is a single faculty teaching each course. The Enrolled relation stores which courses are taken by students, and the grade obtained.

You must create a SQL script for the above tables (place it in the Q1 folder). Also, you have to create two applications (one in C, one in Java) that are run by faculty and students, respectively. Each application must have a command-line interface menu that allows the user to select one option as below. Once that menu function is completed, the program must return to the main menu. For each menu option, you are allowed (and even recommended, if needed) to have multiple steps (or “screens”) to complete the tasks. You will use the DBS2 Oracle instance as DBMS.

Q1: Embedded SQL in C

Faculty Menu:

Application starts by requesting faculty ID, and the remaining session assumes that particular faculty ID is active. If (-1) is introduced, a new faculty is created, and the user is prompted for all necessary information except fid which must be generated by the program. The main menu is the following:

L – List: lists all records in the course table

C – Create: creates an entry in the Courses table; course is taught by the active faculty; user will be prompted for all necessary field values (except cid which is generated by the program); will return error if duplicate cname detected
D – Delete: deletes an entry in the Courses table; only allowed if course (given as input by cid) is taught by active faculty and if there are fewer than 5 enrolled students

S – Search: search course based on substring of course name which is given by user; list all matching courses

M – My Classes: lists all classes taught by the active faculty, and the average student grade for each class

X – Exit: exit application

Q2: JDBC

Student Menu:

Application starts by requesting Student ID, and the remaining session assumes that student ID is active. If -1 is introduced, a new student is created, and the user is prompted for all necessary information (except sid which is generated by the program). The main menu is the following:

L – List: lists all records in the course table

E – Enroll: enrolls the active student in a course; user is prompted for course ID; check for conflicts, i.e., student cannot enroll twice in same course

W – Withdraw: deletes an entry in the Enrolled table corresponding to active student; student is prompted for course ID to be withdrawn from

S – Search: search course based on substring of course name which is given by user; list all matching courses

M – My Classes: lists all classes the active student is enrolled in.

X – Exit: exit application