



COS 221 Practical Assignment 3

- Date Issued: **11th March 2020**
- Date Due: **1st April 2020** before **11:00 AM**
- Submission Procedure: **Upload to the web server (whaetley) and CS web**
- This assignment consists of **5 tasks** for a total of **Total 110 marks, Full marks 110.**

1 Introduction

The university is updating their schedules and your help is needed! A fully-functioning student management system is now required by them. The system should be able to demonstrate modules and timetables management. The system should allow students to register for a module they have prerequisites for and make sure the module is not full (use lecture hall sizes as the maximum number of places available for students). The system should allow a student to be able to create a timetable for the course they are taking as well as allow them to register for a module they want to take. The system should also be able to resolve clashes by providing alternatives or error messages if no alternative is available. The system should also store the final marks of each student per module.

You will be testing your system using the BSc IKS course.

- The venue sizes, lecture times and test times can be downloaded from the student center on your upnet UP portal under the module look up section.
- The list of the courses taken by a BSc IKS student is available on the CS website under the COS000 module in the undergraduate portal.

Your java program should be able to demonstrate these queries

1. Add students to the database. [3]
2. Add venues to the database. [3]
3. Add modules with their prerequisites to the database. [5]
4. Add tests, lectures, exams and supplementary exams to the database. [2]
5. Show schedule for a module(exams, supplementary exams and lectures). [3]
6. Show prerequisites for a module. [3]
7. Show all students registered for a module. [3]
8. Select a student and generate a timetable or display an error if there is clashes. [15]
9. Should not allow a student to register if prerequisites are not met. [10]
10. Should not allow a student to register if there is no space in the module. [8]
11. Should be able to add final marks. [3]
12. Show a list of modules a student has passed. [3]
13. Show a list of modules a student has failed. [3]
14. Show a list of students who need to be academically excluded because they have failed more than 75 percent of their modules. [6]

After successful completion of this assignment you should be able to:

- develop a simple system using java for front-end query creation.
- design a database for any system,
- specify and execute SQL in Java.

2 Constraints

1. You must complete this assignment individually.
2. The system which include, all code developed, database schemas and queries designed and implemented will be marked.
 - (a) Systems which run and perform what they are supposed to do get full marks
 - (b) Systems which run but do not perform as required, will receive partial marks
 - (c) Systems which do not run will be allocated partial marks based on the functionality they would have exhibited.
3. You may ask the Teaching Assistants for help but they will not be able to give you the solutions.
4. You may utilise any text editor or IDE, upon an OS of your choice. In the Informatorium, you will use either MySQL Workbench or MariaDB to create the system database on **wheatley web server**. All files and scripts must be uploaded and run on wheatley for demonstration purposes.
5. You are required to apply the knowledge acquired from the previous lectures by using java and any tool for designing your database schema and ER diagram.
6. **All above mentioned queries should be implemented using a Java program.**
7. **NB: The database must be populated with data that should be able to demonstrate all of the above queries correctly.**
8. **NB: Where possible your queries should return at least 5 results.**

3 Submission Instructions

You are required to upload all your source files and mySQL dump (in an archive) to the Computer Science web-portal. No late submissions will be accepted, so make sure you upload in good time. You will be required to download the files you uploaded to CS web and load them onto **wheatley** as part of the assessment of the practical assignment.

4 Online resources

Access a free SQL Tutorial at: https://www.w3schools.com/sql/sql_create_table.asp

Example to connect to the mysql database in Java: <https://www.javatpoint.com/example-to-connect-to-the-mysql-da>

Example query in Java: <https://alvinalexander.com/java/java-mysql-select-query-example>

There are many other resources online for example Stack overflow – <https://stackoverflow.com/> a platform for developers to learn, share knowledge and build a career.

IMPORTANT NOTE: Make use of your textbook¹ and/or the lecture notes for SQL programming, web programming using Java.

5 Rubric for marking

Database creation	
Tables	5
Columns	5
Keys	5
Database schema populated	10
Using JAVA	
Connecting to your DB via JAVA	5
Program runs and compiles with no errors	5
Query Implementation	
Students added	3
Venues added	3
Modules and prerequisites added	5
Tests etc. added	3
Show schedule for a module.	2
Show prerequisites for a module.	3
Show all students registered for a module.	3
Module selected and timetable generated.	15
Prerequisites not met.	10
No space	8
Should be able to add final marks.	3
List of passed	3
List of failed.	3
Exclusion.	6
Dump	
Correct Database dump	5
Total	110

¹Some of the source code examples and database design techniques are in Edition 7 [?]

6 Assignment Instructions

Task 1: Creating a database (15 marks)

You are required to create a database that will cater completely for the above mentioned queries and scenario. You must include appropriate attributes in your tables.

Task 2: Populating database (10 marks)

The database must be populated with a minimum of 30 students and all the information required for the BSc IKS course. The data should be able to allow each query to return at least 5 results.

Task 3: Java program (10 marks)

Develop a java program that connects to your database and implements all the queries. All query implementation should be done via this program.

Task 4: Query implementation (70 marks)

See above queries for detailed mark description.

Task 5: Database dump (5 marks)

Upload a **Single file** containing all your tables along with the source code of your Java program to the CS website.

IMPORTANT NOTE: Please refer to the rubric for the detailed allocation of marks.