UNIVERSITYOF HERTFORDSHIRE
School of Engineering and Computer Science

COURSEWORK ASSIGNMENT

Module Title: Programming for Software Engineers	**Module Code: 7COM1025 (Semester B)**
Assignment Title: Main Coursework	**Individual Assignment**
Tutor: Dr Hui Cheng	**Internal Moderator**: Dr Raimund Kirner

Student ID Number **ONLY**:	Year Code:

Marks Awarded %:	Marks Awarded after Lateness Penalty applied %:

Penalties for Late Submissions

- For each day (or working day for hard copy* submission only) for up to five days after the published deadline, coursework submitted late (including deferred coursework, but with the exception of referred coursework), will have the numeric grade reduced by 10 grade points until or unless the numeric grade reaches or is 50 (PG).
- Coursework (including deferred coursework) submitted later than five days (five working days in the case of hard copy* submission) after the published deadline will be awarded a grade of zero (0).
- Late submission of referred coursework will automatically be awarded a grade of zero (0).
- Please Note: Work that is submitted through StudyNet (Canvas) is subject strictly to the School policy on late submission – even one second late will be subject to the lateness penalty, with few acceptable reasons for late submissions being acceptable. You are strongly advised to submit your work at least one hour before the submission deadline, to give time to resolve difficulties.

Please refer to your student handbook for details about the grading schemes used by the School when assessing your work. Guidance on assessment will also be given in the Module Guide.

Guidance on avoiding academic assessment offences such as plagiarism and collusion is given at this URL: http://www.studynet.herts.ac.uk/ptl/common/LIS.nsf/lis/citing_menu

ASSIGNMENT BRIEF

Students, you should delete this section before submitting your work.

This Assignment assesses the following module Learning Outcomes (Take these from the module DMD):

- LO4 - Design and implement a substantial software artefact.
- LO5 - Evaluate, choose and employ appropriate data structures and algorithms.
- LO6 - Refactor part of a software item to achieve a specified goal.
- LO7 - Be able to design a test suite and perform a set of tests.

Assignment Brief:

Implement a system according to the following description

The University Sports Centre (USC) needs software for managing the bookings of group exercise classes made by the students. The centre offers different group exercise classes on both Saturday and Sunday. The classes could be Yoga, Zumba, Aquacise, Box Fit, Body Blitz, etc. Each class can accommodate 4 students at most.

For either day (Saturday or Sunday), there are 3 classes per day: 1 in the morning, 1 in the afternoon, 1 in the evening. The price of each class is different. The class price for the same exercise will remain the same even if they run at a different time.

A student who wants to book a class needs to first check the timetable and then select a class on a day. A student can check the timetable by two ways: one is by specifying the date and the other is by specifying the exercise name. Students are allowed to change a booking, provided there are still spaces available for the newly selected class. A student can book as many classes as they want so long as there is no time conflict.

After each group exercise class, students are able to write a review of the class they have attended and provide a numerical rating of the class ranging from 1 to 5 (1: Very dissatisfied, 2: Dissatisfied, 3: Ok, 4: Satisfied, 5: Very Satisfied). The rating information will be recorded in the system.

After four weeks (four weekends), the software system must print:

- a report containing the number of students per group exercise class on each day, along with the average rating;
- a report containing the group exercise which has generated the highest income, counting all the same exercise classes together.

In your final deliverable, provide data covering at least 4 different types of group exercise classes, 10 students, and 20 reviews (with rating), which can then be displayed in the output reports. You need to design at least 8 weekends of timetable (i.e., 48 classes) by yourself.

Note: you do not need any kind of external database for this program. The final program should be self-contained. The reports can be printed to `System.out`, or output to a suitable GUI interface. You also do not need any kind of security protocol. All the students may be assumed to be pre-registered.

Your tasks:

1. Create a class diagram, highlighting the main Java classes in the system and their associations. Include the attributes and key methods.
2. Implement the system, using Junit tests to confirm the functionality.
3. Write a short report (ideally no more than 8 pages) explaining your program.

Discuss:

a) Any assumptions made about the system, in addition to the above description.
b) The overall structure and design of your program.
c) Any design patterns/design principles used.

d) The JUnit tests.
e) Any refactoring used during the development of the system.

Deliverables:

A. Develop your code using bitbucket. Send your username to Hui Cheng by emailing to: h.cheng2@herts.ac.uk, and add “huicheng” to your project.
B. Submit to StudyNet:

1. Your report with the UML class diagrams and snapshots of your version control commit messages in a single PDF document.
2. An executable jar file, demonstrating your final system. The system should have some data preloaded (as described above).
3. A zip-file containing the source code of your project.

Submission Requirements:

Students should submit on StudyNet the following three items:

1. a single pdf document containing the practical report, UML class diagrams, snapshots of version control commit messages,

2. a zipped folder containing the final source code, and

3. an executable jar file for running the final system.

This assignment is worth 50**%** of the overall assessment for this module.

Marks awarded for:

The marks will be based mainly on the report, in the pdf document. Source code will be checked to confirm both that the work has been implemented by yourself, and its design.

. UML Class diagram. (10 marks)

. Working jar file. (40 marks)

. Use of version control, with suitable commit messages. (5 marks)

. Appropriate design and implementation. (30 marks)

. Use of JUnit testing. (10 marks)

. Report format and writing quality. (5 marks)

Total: 100 marks

Typical (hours) required by the student(s) to complete the assignment: 40 hours

Date Work handed out:	**Date Work to be handed in:**	**Target Date for the return of the marked assignment:**
12/02/2020	20/04/2020 23:59	19/05/2020

Type of Feedback to be given for this assignment:

Comments will be provided along with marks.