UNIVERSITYOF HERTFORDSHIRE
School of Engineering and Computer Science

COURSEWORK ASSIGNMENT

Module Title: Programming for Software Engineers	**Module Code: 7COM1025 (Semester B)**
Assignment Title: Ref/Def 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

UNIVERSITYOF HERTFORDSHIRE
School of Engineering and Computer Science

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 all the individual fitness classes bookings made by the customers. There are several coaches who work at the USC. Each coach can teach one or more types of fitness classes, e.g., SPIN, YOGA, BODYSCULPT, ZUMBA, etc. Each customer is registered with one main coach who monitors his/her overall progress. Each customer has a record stored on the system holding their name, gender, DoB, address, emergency contact phone number, main coach and a list of all the individual fitness classes past and booked.

Customers use the software system to book their individual fitness classes. All the classes are one to one and last for one hour. When requesting a class, the customer will specify the needed fitness type (SPIN, YOGA, BODYSCULPT, ZUMBA, etc). The system will then arrange an individual class with date/time convenient to the coach who can teach it and inform the customer. A customer may change the date/time of the booked class subject to the availability of his/her new booking request. A customer may cancel a class, fail to attend a class, or attend a class.

On attending a class, the customer will first be checked in, then will be taught by the booked coach, before finally the class is finished. At the end of the class, the coach may add some notes to the customer's record about progress.

After each month, the University Sports Centre must print a report of how many individual classes each customer has booked, and among them how many were attended, missed or cancelled

In your final deliverable, have the program set up some coaches, customers, and fitness classes in the past and in the future. You should provide data covering at least 3 coaches, 10 customers and 15 fitness classes, which can then be displayed in the output reports.

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 customers and coaches 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: 11/06/2020	**Date Work to be handed in:** 29/06//2019	**Target Date for the return of the marked assignment:** 07/2020

Type of Feedback to be given for this assignment:

Comments will be provided along with marks.