University of Hertfordshire UH

School of Computer of Science
ASSIGNMENT BRIEFING SHEET (2017/18 Academic Year)

Assignment Title	Assignment 1: Practical	**Submission Date**	
Module Title	Programming and Software Engineering Practice	**Module Code**	6WCM0027/6WCM0028
Tutor	Peter Lane	**GROUP or INDIVIDUAL Assignment**	***Individual***

FOR INDIVIDUAL ASSIGNMENTS – *STUDENT TO COMPLETE*

(Comments on this assignment by students can be made on the back of the assignment briefing sheet).
By completing **BOX A** below, I certify that the submitted work is entirely mine and that any material derived or quoted from the published or unpublished work of other persons has been duly acknowledged. **[ref. UPR AS12, section 7 and UPR AS14 (Appendix III)].** I also certify, that any work with human participants has been carried out under an approved ethics protocol in accordance with UPR RE01.
Please print your forename and surname in capitals and provide your ID (srn) number.

BOX A

Student Forename *(in CAPS please)*	**Student Surname** *(in CAPS please)*	**Student ID Number (SRN)**	**Signature of Student**

FOR GROUP ASSIGNMENTS - *STUDENTS TO COMPLETE*

Group Name/Number (if allocated by module team)

(Student comments on this assignment can be made on the back of the assignment briefing sheet)
By completing **BOX B** below, we certify that the submission is entirely ours and that any material derived or quoted from the published or unpublished work of other persons has been duly acknowledged. **[ref. UPR AS/C/6.1, section 7 and UPR AS/C/5 (Appendix III)].)].** We also certify, that any work with human participants has been carried out under an approved ethics protocol in accordance with UPR RE01
*Please print your forenames and surnames in capitals, provide your; - ID numbers, actual time spent on the assignment and your signatures. By signing the submission you certify that this work represents equal contributions from all team members. If this is not the case, the module leader **must** be informed before submission.*

BOX B

Student Forename *(in CAPS please)*	**Student Surname** *(in CAPS please)*	**Student ID Number (SRN)**	**Actual Time Spent by each Student (hours)**	**Signature of Student**

This sheet must be submitted with the assignment, signed and either BOX A or B filled in.
LATE SUBMISSION WILL ATTRACT A STANDARD LATENESS PENALTY.

1. For undergraduate modules, a score of 40% or above represents a pass mark.
2. For postgraduate modules, a score of 50% or above represents a pass mark.
3. For work submitted up to 5 working days late marked is capped to a bare pass (40% for undergraduate and 50% for postgraduate).
4. For work submitted more than 5 working days a mark of zero will be awarded for the assignment.

Regulations governing assessment offences including Plagiarism and Collusion are available from:
http://sitem.herts.ac.uk/secreg/upr/pdf/AS14-Apx3-Assessment%20Offences-v06.0.pdf

University of Hertfordshire UH

School of Computer of Science
ASSIGNMENT BRIEFING SHEET (2017/18 Academic Year)

THE ASSIGNMENT TASK: See below

MODULE LEARNING OUTCOMES ASSESSED BY THIS ASSIGNMENT

1. object-oriented approaches to problem analysis and systems design, their principles and supporting justifications.
2. modelling the requirements of software systems using object-oriented techniques.
3. strategies for evaluating object-oriented software designs and architectures according to established quality criteria.
4. a range of leading software engineering practices.
5. the role of software engineering practices in the development cycle.
6. use appropriate tools, techniques and notations in problem analysis, and in the design of programmed computer systems.
7. select and apply suitable object-oriented tools and techniques productively in the design of software.
8. use and critically evaluate object-oriented design techniques.
9. practice and evaluate a variety of software engineering approaches to developing and evolving software.
10. deploy version control within a software engineering context.

SUBMISSION REQUIREMENTS:

This assignment is to be submitted and marked anonymously. Students should ONLY use their student ID number to identify themselves on their work. Work submitted via StudyNet for anonymous marking will automatically have an anonymity number allocated to it.

Students should submit on studynet the following three items:

1. a single pdf document containing the practical report and UML diagrams,
2. a zipped folder containing the final source code, and
3. an executable jar file for running the final system.

FEEDBACK FROM THIS ASSIGNMENT

Comments will be provided along with marks.

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.

1. UML diagrams	
a. Use cases and use-case diagram	(20 marks)
b. Class diagram	(10 marks)
c. Sequence diagram	(10 marks)
2. Working jar file,	(10 marks)
3. Use of version control, with suitable commit messages.	(5 marks)
4. Appropriate design, implementation and development methodology.	(35 marks)
5. Use of JUnit testing, and quality of tests written.	(10 marks)

Total: 100 marks

Regulations governing assessment offences including Plagiarism and Collusion are available from:
http://sitem.herts.ac.uk/secreg/upr/pdf/AS14-Apx3-Assessment%20Offences-v06.0.pdf

University of Hertfordshire UH

ASSIGNMENT DESCRIPTION

Implement a system according to the following description.

The university is developing a new system to keep track of various personnel, students and other people who contact the university. This system will ultimately be deployed on some tablets in the main reception, and your task is to develop the software that will run on them.

The university system stores information on its teaching staff (“lecturers”) and more senior teaching staff (“professors”). Students also belong to the university (“students”). There are also research staff (“researchers”). For all members of the university, the university maintains a unique ID number, full name, address, and email. Each member of the teaching and research staff has an office, and one or more areas of expertise, such as “security”, “programming”, “data analysis”.

Teaching staff offer classes in their areas of expertise: these classes will be named (for example “An Introduction to Programming”), will take place in a room (e.g. “A15”), and at a specified day/time (e.g. “Monday 14:00-17:00). For simplicity, we assume that classes only take place at a single fixed time and room each week. Rooms have a capacity (e.g. “A15” may only seat 10 students).

Implement a way for students to look up and sign on to classes offered by members of the teaching staff. A student can look up an area of expertise, see a list of teaching staff and the classes they offer, and then sign up for the class they choose. A student may also directly look up a member of teaching staff to find their classes.

Researchers can make appointments with teaching staff, meeting the member of teaching staff in their room at a time they choose.

Business visitors often come to the university. Allow them to look up a researcher or member of teaching staff by expertise or name. Visitors can be given an appointment to see the researcher or teaching staff in their room at a time they choose.

For simplicity, design your system for a single term of 12 weeks. Classes can be assumed to run for the entire 12 weeks. Appointments can be at any time in weeks 1 to 12.

At the end of each month, the university produces a report listing all classes, appointments etc taken by each member of staff. Also, a report will list all classes each student is signed up for.

For a more advanced system, you should check some constraints. Following is a suggested list, but add others that you can think of:

- Students can only sign up for a class where there are spare places
- Students cannot sign up for two classes running at the same time
- Visitors cannot book an appointment when the researcher/member of staff is busy
- Lecturers teach no more than 5 classes in a week
- Professors teach no more than 2 classes in a week

Also advanced, can you include a flexible search system so that students and visitors can find a member of staff even though their full name/details are not known?

Regulations governing assessment offences including Plagiarism and Collusion are available from: http://sitem.herts.ac.uk/secreg/upr/pdf/AS14-Apx3-Assessment%20Offences-v06.0.pdf

In your final deliverable, have the program set up some teaching/research staff, students and classes. Provide data covering at least 5 teaching/research staff, 15 students and 10 classes, which can then be displayed and used for testing.

Note: you do not need any kind of database for this program – use the collection libraries as taught in the module. The final program should be self-contained. Examples of the report output can be printed to System.out, or, when the gui is written, output to a suitable gui widget.

You also do not need any kind of security protocol: just allow the user to identify themselves with a unique name or group.

[INCLUDE THESE NAMES IN YOUR REPORT SO MARKERS CAN USE YOUR SYSTEM]

Your tasks:

1. Create separate use cases for the students, teaching staff, researchers and visitors, and then create one or several UML use-case diagrams (you need to cover all use cases, but may combine them in one diagram, or give separate diagrams for each, as you think is appropriated). You are reminded that use cases are a textual description, which are then represented in the UML diagram.
2. Create a class diagram, highlighting the main classes in the system and their associations. Include the attributes and key operations.
3. Create a sequence diagram showing how a student finds a class with free spaces and books onto the class.
4. Implement the system, using junit tests to confirm the functionality.
5. Add a Swing GUI front-end to the system. Use buttons, panels and/or menus as required to support all functions in the system. You MUST write all the GUI code yourself using the principles given in the module. No marks will be awarded for using any kind of form editor.
6. Write a short report (up to 8 pages should be sufficient) 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. Any refactoring used during the development of the system.
    e. The overall process you followed in developing your software, relating your experiences to unit 5.
    f. The junit tests.

Deliverables:

1. Develop your code using the git version control system and bitbucket. Send your username to Peter Lane, and add uhfoodtutor to your project. Include a screenshot/copy of your commit log within your pdf report.
2. Submit to studynet:
    a. Your 8-page report, UML diagrams and git commit log, in a single PDF document.
    b. An executable jar file, demonstrating your final system. The system should have some data preloaded (as described above).
    c. A zip-file containing the source code of your project.

Regulations governing assessment offences including Plagiarism and Collusion are available from:
http://sitem.herts.ac.uk/secreg/upr/pdf/AS14-Apx3-Assessment%20Offences-v06.0.pdf

University of Hertfordshire UH

DEADLINES AND ASSIGNMENT WEIGHTINGS

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

2 You are expected to spend about 40 Hours to complete this assignment to a satisfactory standard

3 Date assignment set 23/2/2018 Date completed assignment to be handed in 6am 8/5/2018

4 Target date for return of marked assignment 5/6/2018

INTERNAL MODERATION

This assignment has been internally moderated. I confirm: • That the assignment set, meets the requirements of the module and that the brief provides appropriate content for students to successfully complete the assignment. • That the assessment is at an appropriate level and matches QAA level descriptors and is an appropriate form of assessment within the total range of assessments for this module. • That the marking scheme is attached and that students can determine how marks are allocated. • That this assessment can be completed **and** marked within University timeframes, and provides detailed feedback (more than just a grade) that supports learning. .	***Moderator name, signature and date***

Regulations governing assessment offences including Plagiarism and Collusion are available from:
http://sitem.herts.ac.uk/secreg/upr/pdf/AS14-Apx3-Assessment%20Offences-v06.0.pdf