

Assignment Briefing Sheet (2024/25 Academic Year)

Section A: Assignment title, important dates, and weighting

Assignment title:	Coursework	Group or individual:	Individual
Module title:	Programming for Software Engineers	Module code:	7COM1025
Module leader:	Hui Cheng	Moderator's initials:	YZ
Submission deadline:	28-04-2025 9:00 am	Target date for return of marked assignment:	26-05-2025

You are expected to spend about hours to complete this assignment to a satisfactory standard.

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

Notes for students

- For undergraduate modules, a score above 40% represent a pass performance at honours level.
- For postgraduate modules, a score of 50% or above represents a pass mark.
- Late submission of any item of coursework for each day or part thereof (or for hard copy submission only, working day or part thereof) for up to five days after the published deadline, coursework relating to modules at Levels 0, 4, 5, 6 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 40. Where the numeric grade awarded for the assessment is less than 40, no lateness penalty will be applied.
- Late submission of referred coursework will automatically be awarded a grade of zero (0).
- 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).
- Regulations governing assessment offences including Plagiarism and Collusion are available from <https://www.herts.ac.uk/about-us/governance/university-policies-and-regulations-uprs/uprs> (please refer to UPR AS14)
- Guidance on avoiding plagiarism can be found here: https://herts.instructure.com/courses/61421/pages/referencing-avoiding-plagiarism?module_item_id=779436
- Modules may have several components of assessment and may require a pass in all elements. For further details, please consult the relevant Module Handbook (available on Studynet/Canvas, under Module Information) or ask the Module Leader.

Assignment Briefing Sheet (2024/25 Academic Year)

This Assignment assesses the following module Learning Outcomes (from Definitive Module Document):

- 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:

See specification provided on Canvas

Submission Requirements:

Students should submit on Studynet the following three items:

1. a single pdf document containing the report with the UML class diagrams and version control snapshots,
2. a zipped project folder containing the source codes folder, test folder, and an executable jar file for running the final program,
3. a screen recording video showing the functionalities of your final program.

Marks awarded for:

Source code will be checked against the report and the video to confirm both that the work has been implemented by yourself, and its design.

- . UML Class diagram. (10 marks)
- . System functions. (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

Type of Feedback to be given for this assignment:

A detailed mark sheet will be provided; verbal feedback after the marks is released.

Implement a system according to the following description.

The Boost Physio Clinic (BPC) is developing a new booking system for the various treatments it provides. The system stores information on its physiotherapists and patients who take treatments. For all BPC members, the clinic maintains a unique ID number, full name, address, and telephone number. Each physiotherapist has one or more areas of expertise, such as “Physiotherapy”, “Osteopathy”, “Rehabilitation”. Each physiotherapist offers treatments in their areas of expertise. These treatments are named (for example “Neural mobilisation”, “Acupuncture”, “Massage”, “Mobilisation of the spine and joints”, “Pool rehabilitation”).

Each appointment will be booked at a date/time (e.g. “Thursday 1st May 2025, 10:00-11:00) with one physiotherapist. A patient can change their appointment (i.e., cancelling it and booking a new appointment) or simply cancel it before it takes place. An appointment is attended when a patient has been checked in.

Implement the function to add/remove patients. Implement two ways for a patient to look up and book treatment appointments with various physiotherapists.

- 1) A patient may look up an area of expertise, see a list of available physiotherapists and their available treatments including the time, and then book the appointment. For example, two physiotherapists may offer Massage at different times and the patient can book one.
- 2) A patient may also directly look up a physiotherapist by their name to find all available treatments, including the time. Then the patient can book one.

For simplicity, design the treatment timetable for all the physiotherapists for a single term of 4 weeks (within a month). At the end of the term, the clinic produces a report listing all treatment appointments for each physiotherapist, i.e., physiotherapist name, treatment name, patient name, time, and appointment status (booked, cancelled, or attended). The report also needs to print the name list of physiotherapists in descending order of the number of appointments being attended.

In your final program, set up 3-5 physiotherapists, 10-15 patients, and 4-week timetable of the treatments offered by each physiotherapist. These pre-registered data can be hardcoded or read from external text files.

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.

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. Record all the functionalities of your final system in a video.
3. Write a short report (no page limit, about 8 pages should be sufficient) explaining your design and implementation.

Discuss:

- a) Any assumptions made about the system, in addition to the above description.
- b) The overall structure and design of your system.
- c) The JUnit tests.
- d) Any refactoring used during the development of the system.
- e) Any design patterns/design principles used.

Deliverables:

- A. Develop your code using any version control software. Take snapshots of your project repository and commit messages. Include the snapshots in your report.
- B. Submit to Studynet:
 1. Your report with the UML class diagrams and version control snapshots in a single PDF document.
 2. A zipped project folder containing the source codes folder, test folder, and an executable jar file for running the final program.
 3. A screen recording video showing the functionalities of your final program.