Northeastern University
London

Assessment Brief: Individual Coursework 2025–26

Assessment Details

Course Title:	Object-Oriented Design
Course Code:	LCSCI5205
Course Leader:	Waleed Iqbal
Level:	5
First, Second, or Third Sitting:	First
Assessment Title:	Object Oriented design project
Assessment Number:	AE1
Assessment Type:	Set Exercises
Restrictions on Time/Length:	24-32 hours
Assessment Weighting:	70%
Issue Date:	8 September 2025
Hand-in Deadline:	1pm 29 October 2025
Planned Feedback Deadline:	28 calendar days after hand-in deadline or last presentation date
File Format Accepted:	PDF, DOC, DOCX, JPEG, PNG, ZIP
Mode of Submission:	Online (Canvas)
Anonymous Submission:	Yes

Assessment Task

On successful completion of this assignment, you will demonstrate the following:

- The ability to design clear UML class diagrams.
- An understanding of object-oriented programming, including inheritance and interfaces, and demonstrate these concepts through your implementation.

Part 1 - UML Class Diagram Exercise

Design a system using a UML class diagram based on the following description. Imagine you are the owner of a pet store. Design a system using a UML class diagram that will allow you to:

- Maintain a register of your pets, including their prices for customers.
- Maintain a register of pet supplies (food, toys, etc.), including their prices.
- Maintain a register of employees and allow them to log in.
- Maintain a list of returning customers.
- Allow customers to make purchases.
- Implement any other procedures specific to your case.

Implement any other procedures specific to your case.

- Your model must contain:
- At least five classes.
- At least two instances of inheritance.
- At least two interfaces.

The system will use a text interface and keyboard input. Do not worry about the user interface.

Part 2 - Object Oriented Design Project and Object Oriented Coding Exercise

In this part of the assessment exercise, you are asked to develop an application using object-oriented programming to maximise code reuse based on your previous UML diagram.

You will be required to document your code and provide a report detailing how and where your code demonstrates the following features:

- Objects
- Classes
- Encapsulation
- Constructors
- Inheritance
- Polymorphism

For the purpose of this exercise do not worry about the user interface. Your code must be well documented. The program must use standard data structures for example (lists, hash tables, sets, etc.; strings, booleans, integers, floating points)

Discuss with your lecturer any other types of programming applications or ideas for software applications you wish to use.

Submit a report on your program justifying your design choices and UML diagrams. Suggested Tools for UML diagrams rendering include: https://draw.io.

You must also **submit your source code**.

The report and code should be submitted as a .zip file.

Assessment Criteria

Part 1 (30 marks):

- **Design Quality (10 marks):** Marks are awarded for creating a UML class diagram that effectively utilises object-oriented design principles. This includes the justified use of inheritance, abstract classes, polymorphism, and encapsulation.
- **Relationships (10 marks):** Diagrams should clearly represent the relationships and hierarchies between classes.
- **OOP Understanding (10 marks):** Marks are allocated for demonstrating a deep understanding of object-oriented programming through the appropriate application of its features in the diagram. Special attention should be paid to how these features fulfil the specific needs of the proposed system.

Part 2 (60 marks):

- **Code Quality (10 marks):** Marks are awarded for clear, syntactically correct, and well-documented source code.
- **Naming Conventions (10 marks):** proper naming conventions, comment clarity, and the logical organisation of code blocks.
- **Data Structures and Types (10 marks):** Evaluation based on the appropriate and effective use of standard data structures (like lists, dictionaries, sets) and data types (such as strings, integers, booleans, floating points) that support the functionality of the application.
- **Inheritance and Abstraction (10 marks):** Marks are given for correct and innovative use of inheritance and abstract classes to create a well-structured codebase that supports extendibility and reusability.
- **Maintainable Code Structure (10 marks):** Assessment of the code’s organisation, focusing on its maintainability. This includes the use of modular programming practices, clear separation of concerns, and the ability to extend or modify code with minimal impact on existing functionality.

- **Implementation of OOP Concepts (10 marks):** Additional marks for effectively demonstrating other OOP concepts such as polymorphism, access modifiers, static attributes, and static methods in a way that enhances the application's functionality and developer usability.

10 marks are awarded for English proficiency of your report.

Submitting Assessments

You have three submission attempts, but only the last submission will be graded. If your last submission attempt is late, you will receive the late penalty even if you have a previous submission that was on time. Please make sure to avoid multiple submissions for assessments with multiple components, as only the last attempt will be graded. Upload several files in one submission attempt instead.

If your assessment requires anonymous submission (see the assessment details table at the top of your assessment brief), please be sure you have left your name off of your submission and out of the submission file name, as failing to do so may result in a 0% mark on the assessment.

Refer to the assessment details table in your assignment brief for acceptable file formats. Avoid submitting zip files (unless explicitly required by the assessment brief); use the 'add files' function to submit multiple files instead. If you are submitting a physical artefact, you must also provide clear and thorough documentation (such as in the form of photographs or a video) of your submission by the deadline; see the bottom of this section for guidance on submitting video files.

Please ensure that you tick the agreement box at the very bottom of your Canvas submission page (scroll down if you don't see it). This will enable you to select 'Submit Assessment.' Please review the submitted file to ensure that everything is in order.

If you encounter any issues with submission, e-mail a copy of your assignment before the deadline to student.assessments@nulondon.ac.uk along with screenshots of the problem on Canvas, showing a timestamp.

To turn on notifications for submission confirmation emails in your Canvas settings: Account > Notifications > Turn on the bell for 'All submissions.' In the app this is via Settings > Email Notifications > All submissions.

To submit a video recording: Select the 'Panopto video' icon in the text entry box in your submission portal. You can upload a video file of any format from your media library by selecting 'upload,' choosing 'my folder' in the drop down menu, and clicking 'insert.' You should be able to play the video back once it processes. See further explanation, including guidance on recording videos using Panopto, in this support article: 'How to Submit a Video Assignment in Canvas.'

Marking

The University uses two assessment marking schemes – one for undergraduate and one for postgraduate – to mark all taught programmes leading to an award of the University.
More detailed information on the assessment marking scheme and the criteria can be found in the Course Syllabus, available on the University's VLE.

Learning Outcomes

This assessment will enable students to demonstrate in full or in part the learning outcomes identified in the Course Descriptor.

On successful completion of this assessment, students should be able to:

Knowledge and Understanding

K1b Critically understand well-established object-oriented design patterns and software engineering practices.

K3b Choose appropriate object-oriented design patterns and use well-established practices to design small to moderately sized programs.

Subject-Specific Skills

S1b Critically evaluate the impact of program design decisions on the technical, social and management dimensions of software.

S2b Implement a given software design with correct, well-designed, well-written, well-structured, well-tested and well-documented code.

Transferable Skills

T3b Participate as part of a team in the design and implementation of software projects.

T3b Demonstrate a sound technical proficiency in written English and skill in selecting vocabulary so as to communicate effectively to specialist and non-specialist audiences.

Accessing Feedback

Students can expect to receive feedback on all summative coursework within 28 calendar days of the submission deadline or, if applicable, the last oral assessment date, whichever later. The 28 calendar day deadline does not apply to work submitted late. Feedback can be accessed through the assessment link on the Canvas course page.

Late Submissions

Please ensure that you submit your assignment well before the deadline to avoid any late penalties, as a submission made exactly on the deadline will be considered late. Please keep in mind that there may be differences between your computer's clock and the server time, which can cause discrepancies, and that Canvas may take some time to process your submission.

Your Canvas submission portal displays two due dates: one is the deadline for your assignment, and the second is the latest possible date by which your assignment can be submitted late. Please make sure you submit by the assessment deadline in order to avoid late penalties.

If assessments are submitted late without approved Extenuating Circumstances, there are penalties:

- For assessment elements submitted up to one day late, any passing mark will receive 10 marks deducted or a threshold pass (40% for undergraduate students, 50% for postgraduate students), whichever is higher. Any mark below 40% for undergraduate students and below 50% for postgraduate students will stand.
- Students who do not submit their assessment within one day of the deadline, and have no approved Extenuating Circumstances, are deemed not to have submitted and to have failed that assessment element. The mark recorded will be 0%.
- For assessment subelements, late submission will result in non-submission penalties deducted according to the marking criteria above.

For further information, please refer to AQF7 Part C in the Academic Handbook.

Extenuating Circumstances

The University’s Extenuating Circumstances (ECs) procedure is in place if there are genuine circumstances that may prevent a student from submitting an assessment. If the EC application is successful, there will be no academic penalty for missing the published submission deadline.

Students are normally expected to apply for ECs in advance of the assessment deadline. Students may apply for consideration of ECs retrospectively if they can provide evidence that they could not have done so in advance of the deadline. All applications for ECs must be supported by independent evidence.

Successful EC applications for live oral assessments, including vivas, will result in a deferral of the oral to be organised by faculty, students, and Timetabling for a date as close as possible to the original presentation date. The deadline for supplementary materials, if assigned, will be carried forward by the length of the oral assessment extension.

Missing an oral assessment, including a compulsory viva, without an approved EC will result in a non-submission for the entire assessment and, accordingly, a recorded mark of 0%.

Students are reminded that the ECs procedure covers only short-term issues (within 21 days leading to the submission deadline) and that if they experience longer-term matters that impact on learning then they must contact Student Support and Development for advice.

Under the Extenuating Circumstances Policy, students may defer an assessed element on only one occasion and may request an extension on a maximum of two occasions.

For further information, please refer to the Extenuating Circumstances Policy in the Academic Handbook.

Academic Misconduct

Any submission must be a student's own work and, where facts or ideas have been used from other sources, these sources must be appropriately referenced. The University reserves the right to hold a viva if there are concerns about the authenticity of a student's or learner's work. The Academic Misconduct Policy includes the definitions of all practices that will be deemed to constitute academic misconduct. This includes the use of artificial intelligence (AI) where not expressly permitted within the assessment brief, or in a manner other than specified. Students should check this policy before submitting their work. Students suspected of committing Academic Misconduct will face action under the Policy. Where students are found to have committed an offence they will be subject to sanction, which may include failing an assessment, failing a course or being dismissed from the University depending upon the severity of the offence committed. For further information, please refer to the Academic Misconduct Policy in the Academic Handbook.

Version History

Title: Assessment Brief Template **Approved by: The Quality Team**					
Version number	**Date approved**	**Date published**	**Owner**	**Location**	**Proposed next review date**

4.0	March 2023	March 2023	Registrar	VLE/ Faculty Resources Page	March 2024
3.0	August 2022	August 2022	Registrar	VLE, Faculty Resources Page	July 2023
2.3	December 2021	December 2021	Registrar	VLE	August 2022
2.2	August 2021	August 2021	Registrar	VLE	August 2022
2.1	September 2020	September 2020	Registrar	VLE	August 2021
2.0	September 2020	September 2020	Registrar	VLE	August 2021
1.0	August 2019	August 2019	Registrar	VLE	August 2020
Referenced documents	AQF7 Academic Regulations for Taught Awards; Extenuating Circumstances Policy; Academic Misconduct Policy; Course Syllabus				
External Reference Point(s)	UK Quality Code Theme: Assessment				