Assessment Brief: Coursework 2024-25

Assessment Details

Course Title:	Programming Design Paradigm
Course Code:	LCSCI7225
Course Leader:	Jiri Motejlek
Level:	7
Assessment Title:	AE2
Assessment Number:	AE2
Assessment Type:	Coding Assignment
First or Second Sitting:	First
Restrictions on Time/Length:	Code and 3000-word explanation (+/- 10%)
Individual/Group:	Individual
Assessment Weighting:	50%
Issue Date:	16 May 2024
Hand in Deadline:	05 August 2024, 13:00
Feedback Deadline:	28 Calendar Days after hand in deadline
Mode of Submission:	Online
Anonymous Marking:	No

Assessment Task

Develop a simple "Chat Application" integrated with an AI Language Model in Python. This application should demonstrate your understanding and implementation of the Singleton, Observer, and Factory design patterns. Additionally, you are required to produce UML class diagrams of the system.

Chat Application Features:

- Allow users to send and receive messages in real-time.
- Include AI-driven features, such as automated responses or conversation suggestions, utilising an AI Language Model API (API will be provided).

Design Patterns:

- **Singleton**: Ensure there is a single instance managing the AI model's API configuration and access.
- **Observer**: Implement mechanisms to update chat interfaces dynamically as new messages are received or sent.
- **Factory**: Use to handle the creation of various message types, ensuring flexibility and scalability of message management.

UML Diagrams:

- Produce detailed class diagrams outlining the structure of the application, including classes for user management, messaging, and AI integration.
- Include sequence diagrams to depict the interactions for sending a message and receiving an AI-generated response.
- Activity diagrams to show user interactions and system responses are also encouraged.

Code Submission:

- Your Python source code must be submitted, demonstrating the use of the AI Language Model API and the implementation of the design patterns.
- Ensure your code is well-commented to explain critical sections and design decisions.

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 Python source code**.

Submit the report through Canvas, and the code should be uploaded to a Git repository.

Assessment Criteria

[20 marks]

1. Develop an "AI Chat Application" using Python.
2. Utilise the Singleton design pattern to manage the conversation context. This pattern should ensure that there is only one active conversation context globally.

[10 marks]

3. The application connects to the provided AI Language Model API (see supplementary material on Canvas). Demonstrate a successful integration where the application queries the API to retrieve responses based on user inputs.

[10 marks]

4. Implement the Observer design pattern to handle updates in the chat interface. Any new messages or responses should automatically update the chat interface without requiring user intervention to refresh or check for updates.

[20 marks]

5. Incorporate the Factory design pattern to create different types of chat messages or responses. While the application will primarily support text messages, use this pattern to differentiate between user messages, AI responses, and potentially system-generated messages (like errors or notifications).

[15 marks]

6. Write clear and concise code that demonstrates your understanding of each design pattern used. Your code should be well-organised and commented, clearly indicating where and how each design pattern is implemented.

[15 marks]

7. Write a report that includes a UML class diagram, illustrating the design and relationships of the smart home system components.
8. In this report, document your code and provide explanations for each design pattern used.

10 marks are awarded for the English proficiency of your report.

Marking

The University uses two common 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 common 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.

Knowledge and Understanding

K1d Master advanced aspects of software design paradigms.

K2d Master object-oriented solutions to solve small and moderately sized practical problems.

K3d Choose, include, and implement design patterns appropriately to design solutions to problems and understand the impact of the design decisions on the technical, social and management dimensions of software.

Subject-Specific Skills

S1d Implement a given software design with clean, understandable and well-documented code with appropriate unit testing.

S2d Become a sophisticated software developer, familiar with best practices to deliver reusable and extensible code.

S3d Create, refine and express a design in graphical notation such as UML class diagrams.

Transferable Skills

T1d Critically review and defend the design choices made in existing software libraries and frameworks to a group of peers, identify limitations and propose ways for improvement.

T2d Generate appropriate documentation for developed solutions.

T2d Consistently apply an excellent level of technical proficiency in written English, using an advanced application of scholarly terminology, that demonstrates the ability to deal with complex issues both systematically and with sophistication.

T3d Lead or participate in the design and implementation of software artefacts.

Accessing Feedback

Students can expect to receive feedback on all summative coursework within 28 calendar days of the submission deadline. The 28 calendar day deadline does not apply to work submitted late. Feedback can be accessed through the Turnitin assessment link on the course page. Further instructions on submitting an assessment and accessing feedback can be found on the University’s VLE.

Late Submissions

Students are reminded to submit their assessment in the correct format and ahead of the published deadline. Deadlines are strict and Canvas uploads made remotely might not be immediate, we therefore strongly recommend that students upload their work to Canvas in good time before the deadline. If assessments are submitted late without approved Extenuating Circumstances, there are penalties:

- For assessments submitted up to two days late: any mark of 40% or higher will be capped at 40% for undergraduate students. Any mark of 50% or higher will be capped at 50% for postgraduate students. Any mark below 40% for undergraduate students and below 50% for postgraduate students, will stand.
- Students who do not submit their assessment within two days, 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%.
- Late penalties are calculated differently for some types of portfolios. Please read the Assessment Brief of your portfolio carefully.

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 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.

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 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**
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				