University of Lincoln Assessment Framework
Assessment Briefing Template 2024-2025

1. **Module code & title**	CMP1903M – Object Oriented Programming
2. **Assessed learning outcomes**	• [LO1]: Demonstrate the use of version control tools in a software development project • [LO3]: Apply object-oriented principles to the implementation of software programs
3. **Assessment title**	Assessment 1
4. **Contribution to final module mark (%)**	**30%**
5. **Description of assessment task**	This is Assessment 1 and is an **individual** assignment.

This assignment looks at part of the process in implementing a problem – **a code review**. A code review is a review of your code by another developer or developers. Code reviews can help with:

Motivation

Sharing best practice

Also, they can highlight:

Accidental/structural errors

Legibility

Even short, informal code reviews can have a great impact on code quality and error frequency.

You should, for this assignment use, the Github pull request workflow

1.

a) Read the '**Coding Task Guidelines**' in this document carefully.

b) Fork the https://github.com/cfrantzidis/DungeonExplorer repository to your Github account c) Open the repository in Github Desktop and create a 'Development' branch

d) Modify the base code and add functionality by developing your code locally, then committing the changes

e) 'Push' or 'Publish' the changes to your Github repository

f) Create a Pull Request to merge your changes to your main branch

g) Invite two other students to review by adding them as 'Reviewers'. If you cannot do this, you need to add them as 'Collaborators' first (Settings>Collaborators)

h) Once you have received the reviews from two reviewers, make any changes and merge the modified code back into your 'Main' branch.

i) Provide *helpful* reviews on the other students' code

j) Make 2 reviews, receive 2 reviews

k) Complete the self-assessment checklist

2.The questions which you should ask in your review are:

- Have the requirements been met?
- Is the code formatted using the Style Guidelines correctly?
- Is the code easy to read?
- Are different errors handled correctly?

Coding Task Guidelines:

Create the basic game structure for the project "Dungeon Explorer", which involves a text-based adventure game where players navigate through rooms in a dungeon, collect items, battle creatures, and ultimately try to reach a treasure or exit. Here's how it can be structured across the two assignments:

Objective: Create the basic game structure using core object-oriented principles. This phase will focus on defining the game elements, error handling, and basic interactions.

Key Features and Concepts:

1. Classes and Objects:
   - **Game:** Handles the game flow and initializes the player and one simple room.
   - **Player:** Tracks the player's name and a single attribute, such as health or a basic inventory (a single item).
   - **Room:** Represents a single room in the game with a description and possibly an item.
2. Encapsulation and Abstraction
   - Use private fields and provide get/set methods where necessary (e.g., player's health or the room's description).
3. Methods & Constructors
   - Create constructors for the Player and Room classes to initialize their attributes.
   - Implement basic methods:
     - ➢ **Player.PickUpItem():** Adds a single item to the player's inventory.
     - ➢ **Room.GetDescription():** Returns the description of the room.
4. Basic Game Interaction
   - The player starts in a single room and can:
     - ➢ View the room's description.
     - ➢ Display their current status (e.g., inventory and health).
5. Error Checking:
   - Add simple error checking, such as preventing the player from entering a room that does not exist.

6. **Assessment submission instructions**

The submission deadline of this assignment is included in the School Submission dates on Blackboard.

Submit your code repository URL to Blackboard in Assessment 1 Supporting Documents included in the self-assessment checklist which gives you an opportunity to judge your submission.

You should also create:

Create a 3-minute YouTube video reflecting on the code review process, and Object-Oriented features within your code. Details of how to do this are below.

Creating a YouTube video for submission:

1. Create your video. Use screen capture applications such as ScreenPal, OBS, etc.
    a. Use voice over the video to reflect on the code review process and OO programming. In particular:
        i. Point out object instantiation and method use in your code.
        ii. Explain how encapsulation is used in your code.
    b. Please follow the time limitation of the video. If the video is larger than 3 minutes, a penalty criterion would be implemented.
2. Upload to YouTube, setting the video as 'unlisted' – this ensures it doesn't appear in any search listings.

If you are unsure about any aspect of this assessment component, please seek the advice of the module co-ordinator **Dr. Christos Frantzidis** <cfrantzidis@lincoln.ac.uk>

7. **Date for return of mark and feedback**	Please see the **Hand In Dates.xls** spreadsheet. Note: *all marks awarded are provisional until confirmed by the Board of Examiners.*
8. **Feedback format**	Summative feedback will be provided on BlackBoard according to CRG criteria (see CRG file). You will be given formative verbal feedback during the workshop sessions.
9. **Use of Artificial Intelligence (AI) in this assessment**	The use of AI tools to generate all or part of your assessment submission is **not** permitted unless specifically mentioned below.
10. **Marking criteria for assessment**	A Criterion Reference Grid (CRG) is used to evaluate your learning against a set of pre-defined criteria.
11. **Additional information (support, advice, tips etc)**	Students are encouraged to use any lecture and their own personal notes to assist them with the completion of the assessment. Also, students are allowed to use any library and/or online resource as a guide on how to solve the assessment problems. Students are encouraged to seek assistance from any member of the delivery team and particularly from the module coordinator as means to complete the assessment.
12. **Important Information on Dishonesty, Plagiarism and**	University of Lincoln Regulations define plagiarism as *'the passing off of another person's thoughts, ideas, writings or images as one's own...'*. Examples of plagiarism include the unacknowledged use of another person's material whether in original or summary form. Plagiarism also includes the

University of Lincoln
School of Engineering and Physical Sciences
UG Criterion Reference Grid 2024-2025

CMP1903M Object Oriented Programming A02 2024-2025

Learning Outcome	Criterion	Pass	2:2
[LO2] Identify the values of object-oriented design and programming	Illustrate OO features which were used (20%) [Video, Report]	Simple description of the code; fail to mention some OO features which are used in the code. There is some discussion, though, of OO features such as object instantiations and method calls for example.	Clear evidence the code. Inheritance is re where it is evid
[LO3] Apply object-oriented principles to the implementation of software programs	Develop an object-oriented solution to a problem (60%) [Code, Video, Report]	A limited implementation is presented. The application works, however, its functionality is incomplete. For example, the specific requirements (as described in the brief document) are not implemented correctly. Erroneous input is handled but the errors are not handled completely and/or all possible errors are not handled. Some evidence of object-oriented features such as classes, object instantiation and methods/method calls are present, but they may not be implemented well. The checklist and video are completed.	An implementa which works. The player can and view statist The functionali through multipl monsters with to manage an in multiple items. Erroneous input error or excepti All errors may
[LO4] Use testing principles in the testing and debugging of object-oriented applications	Testing practices used to verify/validate a software application (20%) [Code, Video, Report]	A Testing class is used to verify aspects of the game operation. This verification may not be correct though and may not be useful.	A Testing class a Game object are operating co
Weighting is 70% of the module			

Learning Outcome	Criterion	2:1	1
[LO2] Identify the values of object-oriented design and programming	Illustrate OO features which were used (20%) [Video, Report]		Extensive description of the OO features in the code; Inheritance and polymorphism are described and exemplary examples shown in the code. and referred to in the report. An explanation of why they benefit your code is also made. Examples of static OR dynamic polymorphism may be present
[LO3] Apply object-oriented principles to the implementation of software programs	Develop an object-oriented solution to a problem (60%) [Code, Video, Report]		An implementation is presented which is complete. The functionality allows all of the following: 1) navigate through multiple rooms 2) battle monsters with varying difficulty 3) manage an inventory with multiple items. Erroneous input is handled either by error and exception handling. All possible errors are handled. Evidence of additional OO/C# features such as (but not limited to) LINQ, interfaces, virtual/abstract methods are implemented. Protected access control is used. Static(method overloading) and/or Dynamic(method overriding) polymorphism is present The checklist and video are completed.
[LO4] Use testing principles in the testing and debugging of object-oriented applications	Testing practices used to verify/validate a software application (20%) [Code, Video, Report]		A Testing class is used and individual methods are tested. A robust reporting mechanism is used to document the test outcomes (a test log file for example).
Weighting is 70% of the module			

Description of
· Inheritance
Polymorphism

LINQ
Interfaces
Virtual/Abstract Methods

Method Overloading
Method Overwriting

AI Tools	copying of another student's work'. Plagiarism is a serious offence and is treated by the University as a form of academic dishonesty. For more information on examples of Academic Offences, please see the **Academic Offence Guidance**. Please note, if you use AI tools in the production of assessment work **where it is not permitted**, then it will be classed as an academic offence and treated by the University as a form of academic dishonesty. Students are directed to the University Regulations for details of the procedures and penalties involved. For further information, see www.plagiarism.org