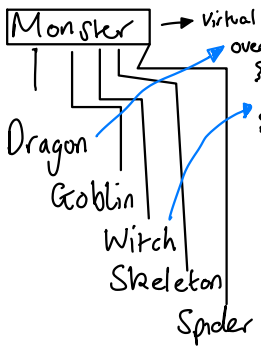
University of Lincoln Assessment Framework
Assessment Briefing Template 2024-2025

1. **Module code & title**	CMP1903M – Object Oriented Programming
2. **Assessed learning outcomes**	• [LO2]: Identify the values of object-oriented design and programming • [LO3]: Apply object-oriented principles to the implementation of software programs • [LO4]: Use testing principles in the testing and debugging of object-oriented applications
3. **Assessment title**	Assessment 2
4. **Contribution to final module mark (%)**	**70%**
5. **Description of assessment task**	This is Assessment 2 and is an **individual** assignment. After a successful first stage interview where you demonstrated use of Git, code review and basic object-oriented principles, you have been asked to demonstrate further object-oriented principles in a second interview: **Coding Task Guidelines:** Expand the "Dungeon Explorer" with advanced OO principles and features: **Key Features and Concepts:** Monsters should have varied difficulty 1. Add new classes: • **Monster:** Represents creatures in rooms. • **Item:** Represents multiple types of items like weapons or potions. • **Inventory:** A collection to manage items. Use a list for varying lengths • **GameMap:** Manages multiple interconnected rooms. → Shows how many rooms left that the user must traverse through 2. Encapsulation and Abstraction • Create hierarchies for: ➢ Creature (abstract class): **Player** and **Monster** inherit from this class. Creature that can hold Weapon Item ➢ Item: Subclasses such as Weapon and Potion. Methods: GetWeapon() 3. Interfaces → IHoldsWeapon Variables: Attack Multiplier • Implement interfaces like IDamageable (applied to both Player and Monster) and ICollectible (applied to items). Creature and child classes are peaceful by default 4. LINQs and Lambda Expressions: • order weapons by damage, • filter between weapon and spell • Use LINQs to filter inventory items (e.g., all weapons) or find the strongest monster in a room. • Use lambda expressions for sorting or filtering. 5. Static and Dynamic Polymorphism: • Implement polymorphic methods:

Common Attributes: Name, Description, Health, SetMaxHealth

Common Attributes: · Name (get/set)



- Different monsters (e.g., Goblin vs. Dragon) have different attack behaviors.
- Items (e.g., Potion vs. Weapon) have unique effects when used.

• Weapon does damage,
• Spell heals the user

6. Error Checking:
   - Enhance error checking for invalid commands or interactions (e.g., trying to use a non-existent item).

Each monster has a varied health and damage multiplier

7. Game Expansion:
   - The player can now:
     - Navigate through multiple rooms. (5 rooms with unique monsters)
     - Battle monsters with varying difficulty.
     - Manage an inventory with multiple items.
       - Multiple weapons and spells

Read the Coding Task Guidelines and perform the following:

- Develop a working solution which showcases your knowledge of the C# language, and object-oriented principles.
- Implement a testing strategy for the solution.
- Create a 5-minute video in which you demonstrate your solution and in particular, its object-oriented features. TA penalty will be applied if you exceed the suggested video duration.
- Fill in the self-reflective assessment of the task using the report template supplied.

8. **Assessment submission instructions**

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

You should submit to Assignment 2 Upload (all in the same document):

a) your code repository URL

b) your YouTube video URL

c) self-reflection of the development exercise

d) the self-assessment checklist.

Creating a YouTube video for submission:

e) Create your video. Use screen capture applications such as ScreenPal, OBS, etc.

a. Use voice over the video to reflect on the dice games and OO programming. In particular:

i. Show how exception handling is used in your code.

ii. Explain how inheritance and polymorphism (if applicable) is used in your code.

f) 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.uk>

9. **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.

10. **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.

11. **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.

12. **Marking criteria for assessment** A Criterion Reference Grid (CRG) is used to evaluate your learning against a set of pre-defined criteria.

13. **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.

14. **Important Information on Dishonesty, Plagiarism and AI Tools** 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 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