



Object Oriented Programming

Hurdle Task 2: Semester Test Resit

Overview

Note: This hurdle task is a time-bound test. You have a 48 hour window during week 12 to complete it.

- **If you receive a Pass grade**, you have passed the hurdle and can include the test as evidence of that in your portfolio.
- **If you receive a Fail grade**, you have not passed the hurdle. The maximum mark you can receive for this unit is 45.

In this unit, you have been using object-oriented programming to implement all of your programs. For this task, you will need to show your understanding of the OO principles.

- Purpose:** Demonstrate your understanding of object-oriented programming and the core concepts of object-oriented design.
- Task:** You must complete two tasks. The first is a coding task, to be submitted as C# source code files and a screenshot showing your program's output. The second task asks for a written response, to be submitted as a PDF.
- Time:** This task should be completed during week 12 — see Canvas for the assignment window.

Submission Details

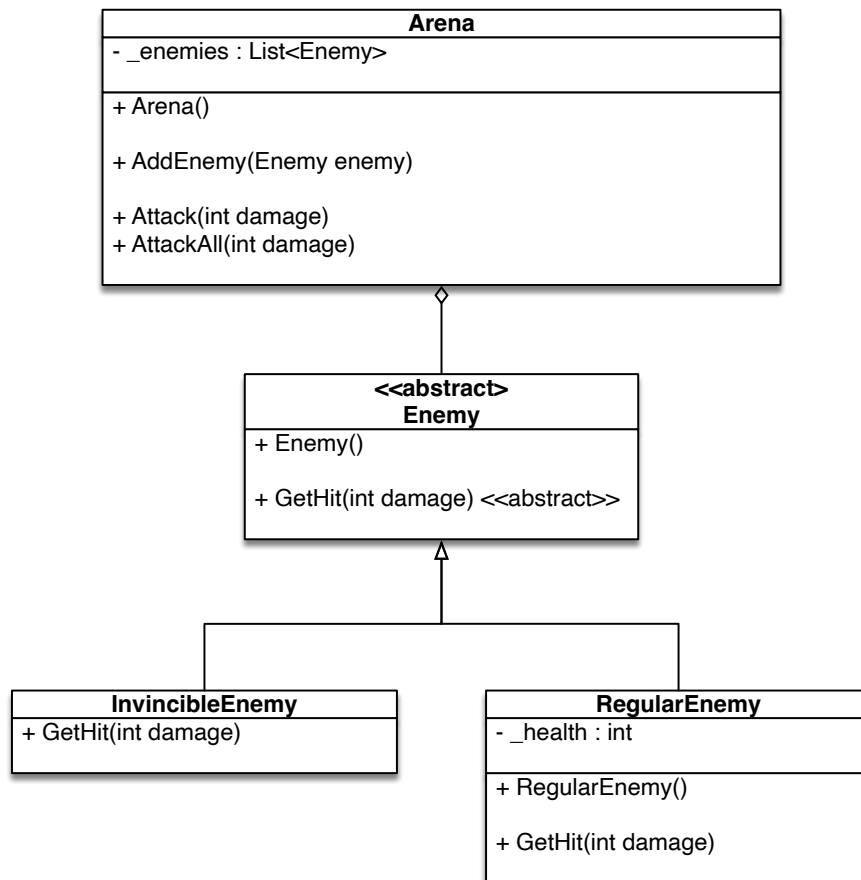
You must submit the following files, formatted using formatmytask.com:

- For Task 1:
 - C# code files of the classes created
 - A screenshot of the program output
- For Task 2:
 - A PDF document with your answer

Make sure that you submit code that is readable and follows the C# conventions.

Task 1

Consider the following program design:



The **Arena** class is responsible for maintaining a list of enemies, and processing any attacks. When created, an **Arena** has no enemies. Enemies can be added to an Arena using the **Add-Enemy** method.

When the **Attack** method is called, one of two things will happen:

- If the Arena contains enemies:
 - “Bring it on!” is printed to the terminal
 - The first enemy in the Arena’s list will **GetHit** for the specified amount of damage
- Else...
 - “Not very effective...” is printed to the terminal

When the **AttackAll** method is called, one of two things will happen:

- If the Arena contains enemies:
 - “Charge!” is printed to the terminal
 - Every enemy in the Arena will **GetHit** for the specified amount of damage

- Else...
 - “There is nobody here...” is printed to the terminal

The **Enemy** abstract class defines anything that can be added to the arena and fought.

The **RegularEnemy** class is a kind of Enemy. It starts with 10 health. When it is hit, one of two things happen:

- If it has more than 0 health:
 - “Ow!” is printed to the terminal
 - It’s health is reduced by the specified amount of damage
- Else...
 - “You already got me!” is printed to the terminal

The **InvincibleEnemy** class is a kind of Enemy. When it is hit, it prints “Ha! Nice try” to the terminal.

Your task is to:

1. Implement the program described above. You must write the code for all classes, and all methods/fields/constructors required.
2. Write a simple Main method to demonstrate your implementation:
 - a) Create an **Arena** object.
 - b) Call the **Attack** method with 5 damage.
 - c) Call the **AttackAll** method with 3 damage.
 - d) Add three **RegularEnemy** objects to the Arena
 - e) Add one **InvincibleEnemy** object to the Arena
 - f) Call the **Attack** method with 10 damage
 - g) Call the **AttackAll** method with 1 damage

Task 2

1. Explain the four principles of object oriented programming. For each of the principles, refer to a piece of work that you have completed for this unit and explain how it demonstrates the principle.

Tip: Do not get distracted by “ad hoc” or “parametric” polymorphism, which are not specific to object oriented programming.

Note: Write your answer **in your own words**. You can use as many reference materials as you like, but you must not directly copy text from anywhere.

Tip: Be concise. You should be able to explain each principle in a single paragraph.

Assessment Criteria

Outcome	Requirements
Pass	Code compiles and runs, and mostly matches the provided design. At least 3 of the 4 core concepts are explained correctly with relevant and correct examples from the unit tasks.
Fail	Pass criteria is not met. OR The submission was not in the correct format.