

COMP319: Software Engineering II

Assignment 1 (2024/2025)

(100% mark for Assignment 1.1 is 20% of COMP319 grade)

Deadline for Assignment 1: 11th of December 2024, 17:00

OBJECTIVE

This assignment Object Oriented Pattern design and involves you producing a class design and implementation for a game.

| | |
|--|---|
| Assignment number | 1 of 1 |
| Weighting | 20% |
| Assignment Circulated date provided to class | 2/10/2024 |
| Deadline Day & Date & Time | 11 th of December 2024 at 17:00 (5 PM) |
| Submission Mode | Electronic submission on Canvas (zip) 1. Understand the key problems driving research and development in contemporary software engineering (eg the need to develop software for embedded systems). 2. Be conversant with approaches to these problems, as well as their advantages, disadvantages, and future research directions. |
| Learning outcome assessed | |
| | |
| Submission necessary in order to satisfy Module requirements | No |

| | |
|--------------------------------|--|
| Purpose of assessment | To assess the students' ability to understand the use of object orientated patterns. |
| Marking criteria | See end of document |
| Late Submission Penalty | Standard UoL Policy |

Instructions

This design task requires you to produce a class design and implementation for a game of space invaders, see Figure 1. Each component of the game shown in the Figure 1 must be modelled to class. There is no need to include sound. The main part of the marking schema is focused on the correct implementation of object patterns, so the actual functioning of the game is only a small percentage of the final marks. All classes included in the final submission should be used, there should be no redundant code.

To help you understand how the game works play it here
<https://freeinvaders.org/>



You are expected to demonstrate the use of the following OO techniques and patterns.

Factory.

Chain of responsibility

Open closed principle

Single responsibility

The code can be written in Java™ or C#.

Responsibilities

You should make a list of the relevant responsibilities for each class and determine what data should be modelled.

Marking

| | |
|--|-----|
| Relevant use of factory class for the problem given | 20% |
| Relevant use of the chain of responsibility for the problem given (2 chains for full marks) | 40% |
| Good open/closed application | 10% |
| Coverage of game functionality and working of game | 10% |
| Single responsibility | 10% |
| Overall code readability and quality | 10% |

Note you must use the patterns given to solve the problem given, so not just have for example a piece of factory code which is to solve a different problem. Please implement the pattern explicitly in your code and not rely of the Java JDK libraries. There is some example code given which show's you how to use some use of the Java image drawing API and how to create the canvas. You can use this code in your final submission. This example code is not properly structured and should NOT be used as an example of good design, it is merely demonstrating the API functionality and the how you can refresh the canvas using a thread.

Submission format

All the source code and relevant resources (e.g. image files) should be zipped up in it correct path structure and uploaded to Canvas.

Manifest file

There should be a manifest file, included in the zip, called manifest.txt which describes each of the Java source files. The format is as follows.

| Name | Purpose | Java pattern of principle included |
|-------------|-----------------------------------|---|
| Screen.java | Displays objects for the game. | Singleton pattern |

Make sure you have highlighted in which source files the aspects of the marking scheme have been covered. Files without a manifest, will lose marks.

The code must compile.