Assignment Brief

Sub

Submit assignment

Marks and feedback

Submission Deadline	Marks and Feedback
Before 10am on: **08/12/2023**	**20 working days after deadline (L4, 5 and 7)** **15 working days after deadline (L6)** **10 working days after deadline (block delivery)** 07/01/2024

Unit title & code	CIS007-3 Comparative Integrated Systems
Assignment number and title	1
Assignment type	Artifact submission (video based on code)
Weighting of assignment	70%
Size or length of assessment	Video no longer than 10 minutes; see below for further requirements
Unit learning outcomes	1. Differentiate and critically question the reasons, context and basic principles of a variety of programming architectures and paradigms relevant to industry standard software solutions. 2. Develop, implement and critically appraise software solutions using different strategies, paradigms and architectures

Completing Your Assignment

What am I required to do in this assignment?

This assignment asks you to reflect on various concepts, paradigms and architectures related to Software Development. Based on your code you have to do a video to discuss the four themes **software design principles (low coupling and high cohesion)**, **event-driven programming, interoperability**, and **virtual identity**. You will develop code in a main-stream programming language chosen by you. Your main submission is the video; but you will also submit supporting evidence that shows that you worked on the code.

Example code will be demonstrated during the practical sessions using Java and Eclipse; there will also be some discussion of JavaScript and PHP.

This is an individual assignment.

The Code:

You are asked to develop a game using the 'Tomato' API - https://marcconrad.com/uob/tomato/doc.php Basic examples are available in Java https://github.com/marcconrad/comparativeintegratedsystems and JavaScript https://marcconrad.com/uob/tomato/ You can develop code based on these examples or do your own code from scratch

(using any programming language). **The final code must be sufficiently complex for you to produce a meaningful video.** For this reason, you should regularly discuss your progress with the tutor during the practical session.

Timeline of the Assignment

<table>
<tr><th>Week</th><th>Tasks</th><th>Software topics covered</th></tr>
<tr><td>1</td><td>Commenting and understanding code design.</td><td>Overview of different programming language, history, application areas.</td></tr>
<tr><td>2</td><td>Introduction to the Assignment</td><td>Low Coupling & High Cohesion</td></tr>
<tr><td>3</td><td rowspan="5">Development of artefact. You are expected to use external sources (such as web services, java packages) and document their use.</td><td>GUI and Events</td></tr>
<tr><td>4</td><td>Interoperability</td></tr>
<tr><td>5</td><td>Authentication & Cookies</td></tr>
<tr><td>6</td><td>Version Control</td></tr>
<tr><td>7</td><td rowspan="4">From Week 7 onward we will cover further topics in the lecture that do not relate directly to the four topics to be covered in your video.</td></tr>
<tr><td>8</td><td>Feedback session with tutor to check progress</td></tr>
<tr><td>9</td><td>Finalize code based on feedback received.</td></tr>
<tr><td>10</td><td>Production of video and additional material.</td></tr>
<tr><td>11</td><td>Assignment Submission</td><td>Revision Session for Exam</td></tr>
<tr><td>12</td><td>Exam Week</td><td>No Lecture</td></tr>
</table>

Deliverables:

1. A video of maximum 10 minutes length where you talk about your code and compare and reflect upon the four themes **software design principles**, **interoperability**, **event-driven programming** and **virtual identity**.
2. Additional material to accompany the video; this must include the full source code of a working implementation. **If you use any code from others (other students or external websites) this must be referenced within the documentation of the source code.** You may also include a transcript of the video, design documentation, evidence of testing and integration of external software. Note that this additional material is not marked directly but serves to inform the video.
3. Evidence that you presented your work to the tutor in Week 8 or earlier.

Please note that this is an individual assignment. Although you can and should collaborate with other students you have to individually produce the video and you are individually responsible for the submission of all the additional material. You must also make clear what code has been written by you and where you used code from others.

In Week 8 you will present your work to the tutor. This session serves to provide evidence that you are actively working on your code.

The grade will be capped by 58 if your submission has one of the following issues. The grade will be capped by 48 if your submission has two or more of the following issues:

- You did not present your code in or before Week 8.

- Code does not address the requirements of the case study.
- No dedicated submission of code (code only visible in video)
- Your video does not have your own voice.

The video must use relevant terminology and be focused around the four themes. It should start with a short demonstration of the working system followed by substantive discussion of the four themes within the context of your code.

Further Notes

Please check BREO regularly for further clarifications and details on the tasks.

Note that copying someone else's code is plagiarism and hence an academic offence. However, the following is allowed and encouraged:

To ask other students for help, to ask for guidance and help in internet forums, use of example code that is available on the internet or in books, use of third-party scripts. **Any such help must be clearly acknowledged and referenced.** Any embedded code which does not originate from you must be clearly marked as such; however, you can freely use the example code provided on the BREO site of the unit. If in doubt, ask your tutor if and how you can use a particular source. References to other software used should be made in the format of comments in your code.

Submissions after the hand-in date will not be accepted unless mitigation has been approved by the University's Student Engagement and Mitigation Team. Please see https://www.beds.ac.uk/student-support/mitigation/ for details.

What do I need to do to pass? (Threshold Expectations from UIF)

- Identify different programming languages, architectures and paradigms within code written by you.
- Use and compare at least two different software architectures.

How do I produce high quality work that merits a good grade?

You produce software that follows a clear rationale that addresses the needs of the case study. The code has been thoroughly tested and is properly commented. It makes meaningful use of externally provided services. The video is clearly focused and uses proper terminology. In the video you discuss confidently your code and consider alternative approaches that you could have taken.

How does this assignment relate to what we are doing in scheduled sessions?

In the lectures we will discuss the various topics that you will implement into code and discuss in the video. The practical sessions provide time and opportunity to discuss your progress with the tutor.

Example code will be provided in Java.

Marks and Feedback

How will my assignment be marked?

Your assignment will be marked according to the threshold expectations and the criteria on the following page.

You can use them to evaluate your own work and consider your grade before you submit.

	3rd Class – 40-49%	Lower 2nd – 50-59%	Upper 2nd – 60-69%	1st Class – 70%+
1	You mention how your code is organized into various components.	You clearly identify the responsibilities of some of the various software components in the code.	You confidently reflect on how your code is structured regarding its various components (such as classes, libraries, packages) and how these address different concerns of your application.	You critically discuss the structure of your code regarding its components (such as, classes, libraries or packages) while considering alternative approaches when implementing your code.
2	You mention event-driven programming and are able to relate it to your code.	You clearly identify various events and their role within your code.	You confidently reflect upon how events are generated in your application and what mechanisms are in place to handle these events.	You justify the approach taken when implementing events and event handler into your software.
3	You mention interoperability and you are able to relate it to the code.	You clearly identify where your code interoperates with someone else's code and what protocol is used to accomplish this.	You confidently reflect on how your code works together with other code written in a different architecture or running on a different system.	You critically justify the approaches you have taken to include third-party software, such as a web service, into your application.
5	You mention virtual identity and are able to relate it to your code.	You clearly identify where you use passwords and / or cookies in your code.	You confidently explain how you have used authentication mechanisms in your code to establish virtual identity.	You critically discuss the authentication mechanisms that are used by your code to establish virtual identity and the role they play to make your application secure while also considering alternative approaches.