Assignment Brief

Assignment submission

Marks and feedback

Assessment deadline	Marks and feedback
To be submitted Before 10 a.m. on:	20 working days after deadline (L3,4, 5,6 and 7)
	15 working days after deadline (block delivery)
05/12/2025	25/12/2025

Please note, for Exams the date is arranged centrally aligned to the academic calendar. Exams timetables will be released 6 weeks before the exam period. Please check the exams calendar regularly on this link (requires BREO login).

Key assignment details	
Unit title & code	CIS046-3 Software For Enterprise
Assignment number & title	1
Assignment type (including exams)	Artifact submission (video based on code)
Weighting of assignment	60%
Size or length of assessment or exam duration	10 minutes (video)
Unit learning outcomes	1. Differentiate and critically question the context and underpinning principles of collectively developing software to industry standard
	2. Develop, implement and critically appraise software solutions that are tailored against specific professional requirements

Assignment Brief Discussion with Students

The assessment brief must be discussed during an in-class session with students within the first 2 weeks of the unit and be accompanied by a screen/podcast on the BREO shell explaining the ***assessment, the rubric and marking criteria. Please confirm the date this discussion is to take place:***

Week 2

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 Computer Science. Based on your code, you will create a video to discuss four key themes: version control, event-driven programming, interoperability, and virtual identity. You will develop code in a mainstream programming language of your choice. Your main submission will be the video; however, you are also required to submit supporting evidence that demonstrates your work on the code.
Example code will be demonstrated during the practical sessions using JavaScript, PHP, and Java. This is an individual assignment.

Example Code:
You may develop a game using the 'Banana' API - https://marcconrad.com/uob/banana/doc.php . Basic examples are available in Java https://github.com/marcconrad/comparativeintegratedsystems and JavaScript https://marcconrad.com/uob/banana/ . You can develop code based on these examples or create your own code from scratch using any programming language. **The final code must be sufficiently complex to support a meaningful video presentation. For this reason, you should regularly discuss your progress with the tutor during practical sessions.**

Timeline of the Assignment

Week	***Tasks***	***Software topics covered***
1	*Commenting and understanding code design.*	*Overview of different programming language, history, application areas.*
2	*Introduction to the Assignment*	Version control
3	*Development of artefact. You are expected to use external sources and document their use.*	*Interoperability*
4		*GUI and Events*
5		*Authentication & Cookies*
6		*From Week 6 onward, we will cover additional topics in the lectures that are not directly related to the four themes you will address in your video.*
7		
8	***Feedback session with tutor to check progress***	
9	*Finalize code based on feedback received.*	
10	*Production of video and additional material.*	
11	***Assignment Submission***	*Revision Session for Exam*

12 & 13	***Exam Week***	*No Lecture*

Deliverables:

1. An authentic video of maximum 10 minutes length where you talk about your code and compare and reflect upon the four themes v**ersion control**, **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, external websites, LLMs) 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 only 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, or earlier, 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. The video should be *authentic*, i.e., it should be obvious that the video reflects your understanding and knowledge of the code and the four themes. Substantive editing of the video is therefore discouraged.

Further Notes

Please check BREO regularly for further clarifications and details on the tasks. Take note of the FAQ that answers several questions about the assignment.

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, code generated by large language models (generative AI). **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? How do I achieve a good grade?

The Unit Information Form states the ***Threshold Expectations*** that inform both you and markers what is the minimum needed to be demonstrated to pass the assessment. They should, therefore, answer the question ***"What do I need to pass?"***. During the assessment introductory session, you should be given the opportunity to check your understanding of the threshold statements and what you need to do to surpass them.

The assessment marking criteria listed below show how your work is assessed. The assessment criteria are informed by the unit's learning outcomes and the assessment task. Carefully reading the assessment criteria should help you understand the aspects that will be used to judge your progress and achievement of the learning outcomes and offer guidance on ***"how do I achieve a good grade".***

Threshold expectations:

- *Implement a distributed architecture for a given scenario*
- *Explain key concepts such as version control, virtual identity, events and interoperability based on the code you have developed.*

How does this assignment relate to 'my learning in this unit and help me develop knowledge and skills that I will need for my future?

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 PHP, JavaScript and Java.

Within the University of Bedfordshire, graduate competencies refer to the skills, knowledge, attributes, and abilities that individuals are expected to possess upon completing their education at the graduate level. Our graduate competencies are the fundamental building blocks in preparing you for the future world of work. They form the foundation for learning outcomes within and beyond your course. They are relevant for every subject, and you can work towards them in different ways, so that you can attain and benefit from them even if you have lots going on outside university. These competencies will be integrated across your course and units.

The table below indicates graduate competencies gained relevant to this assessment.

Graduate competencies; Tick (✔) ones relevant to this assessment (further guidance for staff Graduate competencies | University of Bedfordshire (beds.ac.uk)

Digital Literacy	✔
Collaboration and Communication	✔
Problem solving and critical thinking	✔
Creativity and Entrepreneurship	✔
Adaptability and Resilience	✔
Global Citizenship	
Course specific competencies, e.g., use of external software packages and APIs; software architecture, security aspects.	✔

What should I be aware of when preparing my and how and where should I submit my work?

When submitting work for assessment, you must ensure that the video is all your own work. Failure to do this could result in sanctions, including removal from your course. The Academic Integrity Resource (AIR) has been designed to introduce you to academic integrity and how to demonstrate this during your studies.

The University provides access to Studiosity that connects students with a team of writing specialists who are here to help you with writing and core skills - anytime, anywhere. Studiosity provides you with two online services: Writing Feedback and Connect Live. Their Writing Feedback gives feedback on referencing, choice of language, structure and spelling/grammar within 24 hours. If you are not sure how to use the feedback to enhance your work before making a final submission, you can arrange to discuss this with a member of the Study Hub team. Connect Live also offers live support in areas such as basic English, maths and stats, science and generic study skills. Both services are available 24/7, 365 days of the year, and you can access them for FREE via BREO.

For exams, you should familiarise yourself with the guidance information available here: https://www.beds.ac.uk/exams/
Practice exams can be taken for all exam types before exams week. Please see section Prepare for your exam.

You are allowed to use generative AI for the code of this assignment, but not for the video.

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, code generated by large language models. **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.

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.

	40-49% **(3rd Class)** **Threshold Standard**	**50-59%** **(2:2)**	**60-69%** **(2:1)**	**70%+** **(1st Class)**
1	You mention the basic use of version control tools, such as Git, for tracking changes in your code. You also identify that your code is organized into various components.	You clearly identify how version control supports the responsibilities of software components in your project. You reflect on how organized code (classes, libraries, or packages) aids collaboration and efficient management within a version-controlled environment.	You confidently reflect on how the organization of your code and the use of version control address key concerns such as modularity, maintainability, and collaboration. You discuss specific practices like branch management, commit granularity, and using version control to manage dependencies between components.	You critically evaluate the structure of your code and its organization into components, considering alternative approaches. You also reflect on how version control strategies, like branching, tagging, and code reviews, enhance the quality, scalability, and maintainability of your application.
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.
4	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.