# **ASSESSMENT 1 BRIEF**

|  |  |
| --- | --- |
| *MODULE CODE* | COM5003 |
| *MODULE TITLE* | Further Software Development |
| *MODULE LEADER* | Dr Antesar Shabut |
| *ASSESSMENT TITLE* | Case Study |
| *WEIGHTING* | 50% |

## ASSESSMENT LEARNING OUTCOMES

Upon successful completion of this assessment, you will be able to:

1. Demonstrate capacity to code effectively in an object-oriented language to meet given specifications.
2. Select and apply appropriate data structures and algorithms to a given problem.
3. Critically evaluate a computer program regarding robustness, usability, security, maintainability, readability and efficiency considering OOP principles.
4. Test and evaluate the efficiency of an implementation relative to a given problem.

## INSTRUCTIONS

This assessment has been designed to provide you with an opportunity to apply your software development skills in the context of Object-Oriented Programming principles to solve a real-world case study. Your task is to design and implement a solution for the business problem described on **page 2**, employing Design principles and Unified Modelling Language (UML), specifically utilising UML class diagram notations. You need to the following:

* Work with the Product Owner to fully understand the project requirements, ensuring all specifications are clear. Make sure to ask any necessary questions.
* Design your code using a class diagram, applying object-oriented programming (OOP) principles, and establishing appropriate relationships between classes.
* Plan your database structure and select appropriate data structures based on the case study requirements.
* Develop your application by implementing the code and designing the graphical user interface (GUI). Ensure the application is functional and efficient by performing unit testing and following test-driven development (TDD) methodology.
* Evaluate your code in terms of robustness, usability, security, maintainability, readability, and scalability. Highlight the key strengths and selling points of your application.
* Create a video (maximum 5 minutes) showcasing the final application features.

## SUBMISSION DETAILS

|  |  |
| --- | --- |
| *RELEASE DATE* | 28 October 2024 |
| *SUBMISSION DATE* | 13 December 2024 |
| *DELIVERABLES* | Your submission should include the following components:   * Zipped Java code of your application: Submit a zipped folder containing all the code files required to run your application. * Word document (.DOCX): This document should include: * A link to your GitHub repository. * A ~700-word reflection that explains your design and outlines your plan for ensuring the long-term resilience of your application. * A link to a media file (.MP4) that demonstrates the final application and highlights its key features. * Evidence of completing lab tasks: Provide proof of completion for all lab tasks by organising them in the same GitHub repository. Arrange the files into clearly labelled folders (e.g., *Assignment\_1*, *Session\_1*, *Session\_2*, etc.). |
| *SUBMISSION DETAILS* | Submit your assignment by uploading it to Moodle before midday on the submission date. This deadline will be automatically and strictly enforced. If your submission is late, your grade may be affected. If you have any issues submitting your work, you must email the assessment team and copy in the module leader before the assessment due time. Do not leave your submission until the last minute to avoid any penalties due to problems with the submission portal.  Assessment Team: [assessment@leedstrinity.ac.uk](mailto:assessment@leedstrinity.ac.uk)  Module Leader: a.shabut[@leedstrinity.ac.uk](mailto:n.danino@leedstrinity.ac.uk)  We may ask for a demonstration of your work following the submission. If needed, this will be communicated to you individually via email. Please check your emails regularly. |

Your feedback / feed forward and mark for this assessment will be provided within 15 working days.

**Problem Statement: Degree Classification Calculator App**

As university students near the completion of their degree, they often seek to estimate their final classification based on the grades they have earned so far, along with predicted grades for ongoing or upcoming modules. This helps them assess their academic standing and set realistic goals for improvement.

The task is to develop a Degree Classification Calculator App that will enable students to calculate their estimated degree classification based on actual or predicted module grades. The app should support the typical classifications used in degree programmes (e.g., First Class, Upper Second Class, etc.) and provide users with an easy-to-use interface to input their grades. In this project, we will only look at how to calculate UG degree.

Key Features:

* Allow students to input actual grades and/or predicted grades.
* Calculate the overall grade average and display the estimated degree classification.
* Offer flexibility for different grading weightings (use the document attached to resources to understand various ways of weightings).
* Display a clear summary of the results and classification, with an option to adjust input for "what if" scenarios.
* Include an administrative scenario in the use cases to enhance the overall functionality and user experience.

The app should be user-friendly, reliable, and designed to help students better understand their academic standing and set personal academic goals.

Resources:

* Link to the current university degree classification app: [DegreeCalcAIO-v1 - Power Apps](https://apps.powerapps.com/play/e/default-df4c20ba-64a8-4352-b3f9-47881abbc09a/a/9b2a2da9-0535-44c6-9048-bb3918c745fc?tenantId=df4c20ba-64a8-4352-b3f9-47881abbc09a&hidenavbar=true).
* Link to understanding how to calculate your Undergraduate degree document: [Undergraduate Degree Calculations Explained.pdf (sharepoint.com)](https://leedstrinity.sharepoint.com/myLTU/Esssential%20Info/Undergraduate%20Degree%20Calculations%20Explained.pdf).
* The “Calculating Degree” section of [myLTU](https://myltu.leedstrinity.ac.uk/campusm/home#menu), the [Degree Outcomes Statement](https://www.leedstrinity.ac.uk/media/site-assets/documents/key-documents/pdfs/degree-outcomes-statement.pdf) (particularly section “3. Classification Algorithms”)
* The [Taught Programme Academic Regulations](https://www.leedstrinity.ac.uk/media/site-assets/documents/key-documents/pdfs/taught-programme-academic-regulations.pdf) (particularly, paragraph 27 onwards of the “Award” section, of the “Bachelor’s Degree Programme Regulations”).

You will be provided with supporting documents to help you understand the degree classification calculation. Additionally, a meeting with the product owner, Chris Rowley, Assessment Administrator, will be arranged to clarify any questions and provide further guidance on the calculation process.

## MARKING CRITERIA

Marks are awarded based on the following criteria. Within each part, aim to complete the work for each section before moving on to the next. The following banded marking scheme is used:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Exceptional 1st* | 100/95/92 |  | *2:ii* | 58/55/52 |
| *Outstanding 1st* | 88/85/82 |  | *3rd* | 48/45/42 |
| *1st* | 78/75/72 |  | *Bare Fail* | 38/35/32 |
| *2:i* | 68/65/62 |  | *Fail* | 25/20/10/0 |

If you have completed all the preparatory exercises and attended your classes, the estimated additional time required to PASS this assessment is approximately 40 hours.

|  |  |
| --- | --- |
| To obtain a 3rd mark (40%), you must have: | * Demonstrated a basic understanding of OOP code design and implementation principles. * Produced an app that functions at a basic level. * Demonstrated minimal understanding of class types and interactions between classes. * Produced code with some detail and documentation. * Demonstrated some understanding of data structures and data bases. * Demonstrated some understanding of unit testing and TDD. * Produced a report with some detail. |
| To obtain a 2:ii mark (50%), you must have (in addition to the above): | * Demonstrated an understanding of OOP code design and implementation principles. * Produced a functional app with some interactive features. * Demonstrated an understanding of class types and interactions between classes. * Produced code with moderate detail and documentation. * Demonstrated some understanding of data structures and data bases. * Demonstrated some understanding of unit testing and TDD principles. * Produced a report covering key aspects. |
| To obtain a 2:i mark (60%), you must have (in addition to the above): | * Implemented all features from the problem statement. * Demonstrated a solid understanding of OOP code design and implementation principles. * Produced an app that functions well with several interactive features. * Demonstrated good understanding of class types and interactions between classes. * Demonstrated good understanding of data structures and data bases. * Demonstrated good understanding of unit testing and TDD principles. * Produced a reflective report with clear results, discussion and brand resilience plan. |
| To obtain a 1st mark (70%), you must have (in addition to the above): | * Demonstrated a thorough understanding of OOP code design and implementation principles. * Produced an app that functions effectively with all interactive features fully working. * Demonstrated good understanding of class types and interactions between classes. * Demonstrated a thorough understanding of data structures and data bases. * Demonstrated a comprehensive understanding of unit testing and TDD principles. * Produced a reflective report with effective results, discussion and brand resilience plan. |

## ACADEMIC MISCONDUCT

Academic Misconduct includes all forms of academic dishonesty, whether intentional or accidental, that compromise the integrity of the University’s assessment processes. It is essential that you review our [Student Academic Misconduct Policy](https://leedstrinity.sharepoint.com/:b:/r/sites/AcademicQualityOffice/Academic%20Misconduct/Student%20Academic%20Misconduct%20Policy.pdf?csf=1&web=1&e=nzRFVf) to understand the guidelines and the serious consequences that may arise if they are not followed.

## HELP AND SUPPORT

* Please use the module handbook and the [School of Computer Science Community Teams site](https://teams.microsoft.com/l/team/19%3AQlYY5-Mv9kd7wLbFkfElaWi_CyFRV9H0zn42tc321MM1%40thread.tacv2/conversations) as a source of information. Do try and find the answer out yourself before reaching out for help.
* Support will be provided via Microsoft Teams and email during office working hours. You can also ask questions during your timetabled sessions. You may request a one-to-one meeting with a tutor during their published office hours.
* The Student Support team are available seven days a week to support you in all aspects of student life. This could be for support relating to your course, your accommodation or for more general advice such as relationships or your wellbeing. Log in to the LTU app to access support services.
* The full set of university guidelines on assessments, deadlines, and extensions is available on the LTU app, please familiarise yourself with the documentation.