

# Assessment Brief - Coursework

Academic Year	2025/2026
Semester	1
Module Number	CM2601
Module Title	Object Oriented Development
Assessment Method	Individual Coursework
Deadline (time and date)	Nov 2025
Submission	<b>2 Assessment Dropboxes in the Module Study Area in CampusMoodle.</b> <ol style="list-style-type: none"> <li><b>Dropbox 1 - Application</b></li> <li><b>Dropbox 2 - Report</b></li> </ol>
Word Limit	1500
Use of Generative Artificial Intelligence (AI) text	IS NOT authorised
Module Co-ordinator	Malsha Fernando
<b>What knowledge and/or skills will I develop by undertaking the assessment?</b> <b>Develop programming code using the Object-Oriented paradigm (OO) and to apply OO concepts to model and implement a software solution for a given real-world problem.</b>  You will be able to; <ol style="list-style-type: none"> <li>Use Classes and objects: attributes and operations, constructors, access specifiers and non-access specifiers, return types.</li> <li>Apply operator overloading and conversions using object-oriented programming principles: encapsulation, inheritance and polymorphism and abstraction.</li> <li>Use I/O streams to perform file handling with advanced programming features: exception-handling mechanisms, Database connectivity and concurrent programming mechanisms.</li> <li>Design object-oriented solutions for real-world problems using Unified Modelling Language</li> </ol>	

<b>Academic Year</b>	<b>2025/2026</b>
<p><b>On successful completion of the assessment students will be able to achieve the following Learning Outcomes:</b></p> <ol style="list-style-type: none"> <li>1. Analyse a programming task using OO concepts.</li> <li>2. Design software using UML diagrams associated with OO concepts.</li> <li>3. Engineer software guided by UML diagrams and including advanced OO constructs.</li> <li>4. Implement and evaluate an OO software artefact for a given real-world programming problem.</li> </ol>	
<p><b>Please also refer to the Module Descriptor, available from the module Moodle study area.</b></p>	

<b>What is expected of me in this assessment?</b>
<p><b>Task(s) – content</b></p> <p><b>TeamMate: Intelligent Team Formation System for University Gaming Club</b></p> <p><b>Overview:</b></p> <p>Design and implement a Java-based application that helps a <b>university gaming club</b> to automatically form <b>balanced teams</b> for upcoming <b>tournaments, friendly matches, or inter-university events</b>.</p> <p>The system gathers data from club members through a <b>quick personality and interest survey</b>, and then forms <b>diverse, well-rounded teams</b> based on:</p> <ul style="list-style-type: none"> <li>• <b>Game/Sport type</b></li> <li>• <b>Skill level</b></li> <li>• <b>Preferred playing role</b></li> <li>• <b>Personality traits</b></li> </ul> <p><b>Key Features of the Desired Solution:</b></p> <ol style="list-style-type: none"> <li>1. Input &amp; Survey <ol style="list-style-type: none"> <li>a. Members complete a short survey:</li> <li>b. 5 personality questions (check starter pack)</li> <li>c. Interest selection: (e.g., Valorant, Dota, FIFA, Basketball, Badminton)</li> <li>d. Preferred playing role (check starter pack)</li> </ol> </li> <li>2. Personality Types (based on total score): <ol style="list-style-type: none"> <li>a. Leader: 90–100</li> <li>b. Balanced: 70–89</li> </ol> </li> </ol>

## What is expected of me in this assessment?

- c. Thinker: 50–69
- 3. Matching Algorithm (check starter pack): The system builds balanced teams of size  $N$ , ensuring:
  - i. Diverse interests (different games/sports)
  - ii. Role variety (e.g., at least 1 defender, 1 strategist, etc.)
  - iii. Mixed personality types for strong team dynamics
- 4. File Handling
  - a. Load data from a sample CSV (provided in starter pack)
  - b. Save formed teams into another CSV (e.g., `formed_teams.csv`)
- 5. Exception Handling
  - a. Handle missing/invalid inputs
  - b. Validate role and personality scores
  - c. Catch file read/write errors
- 6. Concurrency: Use threads for:
  - a. Processing survey data
  - b. Forming teams in parallel (especially for large datasets)

## Project Tasks

- 1. Design
  - a. Use Case Diagram and Descriptions: Actors may include Organizer (Uploads CSV, defines team size, initiates team formation) and Participant (Completes survey, provides role & skill info).
  - b. Activity Diagrams for each Use case, identifying main flow and alternate flows (error handling).
  - c. Class Diagram including classes like, Participant, Team, TeamBuilder, PersonalityClassifier, etc. Clearly identify the relationships between the Entity classes ONLY.
  - d. Sequence Diagrams for the Use Cases clearly identifying the messages passed between classes and objects.
- 2. Implementation
  - a. Core Logic: Start by designing the class structure and setting up basic functionality, identify suitable logic to classify personality types based on score and match users into balanced teams of size  $N$

## What is expected of me in this assessment?

- b. File Handling and DB handling (if required – not mandatory)
- c. Exception Handling: Implement checks and error-handling mechanisms for file access, invalid inputs, concurrency issues, etc.

### Project Timeline (7 – 8 weeks):

1. Week 1-2: UML Design and Core Structure (Use case, class, activity, sequence diagrams).
2. Week 2-5: Implement key features. File Handling and Exception Handling. Set up concurrency functions.
3. Week 5: Integrate modules.
4. Week 6-7: Finalize all functionalities, add logging, error handling, and improve UI (console-based or basic GUI).
5. Week 7-8: Comprehensive testing (unit testing, concurrency testing, file integrity checks, user acceptance testing, etc.).

*Starter pack link available [here](#).*

### Submission Format

1. You must provide a **single Java Application** in compressed format (.zip) for the above scenario with a **single Report** (.pdf) comprising of requirement analysis, design, and testing.
2. The application must include all project folders/packages/ files required to successfully execute the application on any desktop device using Java.
3. Make sure to include **screen captures of the version controlling logs from start to end** (timestamps must be clear) of the project as an appendix.
4. The report must include following chapters.
  - a) Introduction – Problem identification and solution with timeline, and a brief on the scope of the project.
  - b) Requirement analysis – Stakeholder identification with requirements and Use case diagram with the descriptions.
  - c) Activity diagrams
  - d) A single class diagram
  - e) Sequence diagrams
  - f) Testing and evaluation

## What is expected of me in this assessment?

- g) References
- h) Appendix – Version controlling evidence of logs

### Submission and Viva-voce examination

1. Make sure to **submit by the due date**.
2. **Two separate Dropboxes** will be available on CampusMoodle, i.e. for Application and for Report.
3. Make sure to **clearly indicate your RGU ID and name** in each submission sub file/folder.
4. The final submission (application and report) must be **titled using ONLY the RGU ID**.
5. Not adhering to correct naming requirements stated above will result in a penalty.
6. The demonstration/presentation of the coursework followed by a **physical viva-voce examination** will be conducted one week following the submission.
7. The schedule for viva will be posted on CampusMoodle after the submission deadline.
8. Participation for the viva voce is mandatory. Failing to appear will result in a NS grade for coursework component.

## How will I be graded?

A number of subgrades will be provided for each criterion on the feedback grid which is specific to the assessment.

The overall grade for the assessment will be calculated using the algorithm below\*.

<b>A</b>	At least 50% of the subgrades to be at Grade A, at least 80% of the subgrades to be at Grade B or better, and normally 100% of the subgrades to be at Grade C or better.
<b>B</b>	At least 50% of the subgrades to be at Grade B or better, at least 80% of the subgrades to be at Grade C or better, and normally 100% of the subgrades to be at Grade D or better.

How will I be graded?	
<b>C</b>	At least 50% of the subgrades to be at Grade C or better, and at least 80% of the subgrades to be at Grade D or better.
<b>D</b>	At least 50% of the subgrades to be at Grade D or better, and at least 80% of the subgrades to be at Grade E or better.
<b>E</b>	At least 50% of the subgrades to be at Grade E or better.
<b>F</b>	Failing to achieve at least 50% of the subgrades to be at Grade E or better.
<b>NS</b>	Non-submission.

\*If the word count is above the specified word limit by more than 10% or the submission contains an excessive use of text within tables, the grade for the submission will be reduced to the next lowest grade.

## Feedback grid

GRADE	A	B	C	D	E	F
DEFINITION / CRITERIA (WEIGHTING)	EXCELLENT Outstanding Performance	COMMENDABLE/VERY GOOD Meritorious Performance	GOOD Highly Competent Performance	SATISFACTORY Competent Performance	BORDERLINE FAIL	UNSATISFACTORY Fail
<b>Requirement Analysis and Design (3 subgrades)</b>	Excellent identification of all stakeholders of the systems with requirements and mapping them into external users in the use case diagram. Correct identification of use cases and relationships between use cases. Excellent and correct flow of events presented in use case descriptions. Excellent identification of the correct logical flow of tasks in the activity diagram. Identification of all required classes with correct relationships. Correct mapping and identification of communication between classes/objects. Correct use of notations.	Very good identification of all most all stakeholders of the systems with requirements and mapping them into external users in the use case diagram. Correct identification of use cases and relationships between use cases. Correct flow of events presented in use case descriptions. Excellent identification of the correct logical flow of tasks in the activity diagram. Identification of all most all required classes with correct relationships. Correct mapping and identification of communication between classes/objects. Correct use of notations.	Good identification of most stakeholders of the systems with requirements and mapping them into external users in the use case diagram. Correct identification of use cases and relationships between use cases. Correct flow of events presented in use case descriptions. Good identification of the logical flow of tasks in the activity diagram. Identification of most required classes with relationships. Correct mapping and identification of communication between classes/objects. Improvements can be made in identifying more suitable flow of events and more relevant classes with relationships. Correct use of notations in most diagrams, improvements can be made.	Some identification of stakeholders of the systems with requirements and mapping them into external users in the use case diagram. Identification of use cases and some relationships between use cases. Flow of events presented in use case descriptions. Identification of the logical flow of tasks in the activity diagram. Identification of some required classes with relationships. Some identification of communication between classes/objects. Improvements can be made in identifying more suitable flow of events and more relevant classes with relationships. Errors in the use of UML notations.	Erroneous identification of stakeholders of the systems with requirements and mapping them into external users in the use case diagram. Identification of a few use cases and no relationships between use cases. Flow of events presented in use case descriptions are erroneous. Some identification of the logical flow of tasks in the activity diagram. Identification of some required classes with relationships, but the errors. Some identification of communication between classes/objects. Missing diagrams. Improvements must be made in all most all aspects of the designs, i.e., identification of requirements, mapping requirements to design and use correct UML notations.	The presented designs do not map the UML standards. Incorrect identification and mapping of requirements to design. Major improvements suggested in all aspects of the design. Or no design provided.
<b>Implementation (3 subgrades)</b>	Excellent mapping of the designs to code incorporating all necessary OOP concepts. Good integration of all functional units into a single application. Good coding practices such as commenting, version controlling and adherence to naming standards followed throughout the application. Good level of exceptional/error handling with input validations for all required functions. Correct formatted output with clear instructions provided to the end-user.	Very good mapping of the designs to code incorporating all necessary OOP concepts. Good coding practices such as commenting, version controlling and adherence to naming standards followed throughout the application. Good integration of all functional units into a single application. Good level of exceptional/error handling with input validations for some functions. Correct formatted output with instructions provided to the end-user.	Fair level of mapping the designs to code incorporating some of necessary OOP concepts. Some good coding practices such as commenting, version controlling and adherence to naming standards followed but not all met throughout the application. Some integration of functional units into a single application. Some level of exceptional/error handling without input validations. Correct output provided. Improvements can be made so that the application incorporates all necessary OOP concepts. Minor improvements can be made when mapping the designs to code, making sure to maintain the correct logical flow.	Some level of mapping the designs to code incorporating some of necessary OOP concepts. Only a limited or no good coding practices such as commenting, version controlling and adherence to naming standards followed. No or partial integration of functional units into a single application. No exception handling and no input validations. Correct output provided. Improvements can be made so that the application incorporates all necessary OOP concepts. Improvements must be made when mapping the designs to code, making sure to maintain the correct logical flow.	No level of mapping the designs to code. Some of necessary OOP concepts used with errors. Only a limited or no good coding practices such as commenting, version controlling and adherence to naming standards followed. No or partial integration of functional units into a single application. No exception handling and no input validations. Erroneous output but the application executes. Major improvements must be made so that the application incorporates all necessary OOP concepts. Improvements must be made when mapping the designs to code, making sure to maintain the correct logical flow.	No implementation provided or major errors in the application which does not execute as expected.

GRADE	A	B	C	D	E	F
DEFINITION / CRITERIA (WEIGHTING)	EXCELLENT Outstanding Performance	COMMENDABLE/VERY GOOD Meritorious Performance	GOOD Highly Competent Performance	SATISFACTORY Competent Performance	BORDERLINE FAIL	UNSATISFACTORY Fail
<b>Testing and Documentation (2 subgrades)</b>	Excellent test plan covering all aspects of the project and functions. Good documentation provided which follows formal report writing style and formatting.	Good test plan covering all aspects of the project and functions. Good documentation provided which follows formal report writing style and formatting.	Test plan covering most aspects of the project and functions. Fair level of documentation provided which some what follows formal report writing style and formatting.	Some test plan covering some aspects of the project and functions. Documentation provided which does not follow formal report writing style and formatting.	Some test plan covering only a few aspects of the project and functions. Document does not follow formal report writing standards. Missing sections and formatting.	No test plans. Missing sections and formatting or no document provided.
<b>Demonstration and Viva (2 subgrades)</b>	The presentation and demonstration of skill is outstanding in terms of the scholarly style, clarity, relevance and coverage of the material, knowledge and skill, formatting, referencing, grammar, sentence structure, punctuation, and word choice.	The presentation and demonstration of skill is very good in terms of the scholarly style, clarity, relevance and coverage of the material, knowledge and skill, formatting, referencing, grammar, sentence structure, punctuation, and word choice.	The presentation and demonstration of skill is good in terms of the scholarly style, clarity, relevance and coverage of the material, knowledge and skill, formatting, referencing, grammar, sentence structure, punctuation, and word choice.	The presentation and demonstration of skill is satisfactory in terms of the scholarly style, clarity, relevance and coverage of the material, knowledge and skill, formatting, referencing, grammar, sentence structure, punctuation, and word choice.	The presentation and demonstration of skill is weak and needs improvement in terms of the scholarly style, clarity, relevance and coverage of the material, knowledge and skill, formatting, referencing, grammar, sentence structure, punctuation, and word choice.	The presentation and demonstration of skill is not satisfactory in terms of the scholarly style, clarity, relevance and coverage of the material, knowledge and skill, formatting, referencing, grammar, sentence structure, punctuation, and word choice.

***Coursework received late, without valid reason, will be regarded as a non-submission (NS) and one of your assessment opportunities will be lost.***



## What else is important to my assessment?

### What is the Assessment Word Limit Statement?

It is important that you adhere to the Word Limit specified above. The Assessment Word Limit Statement can be found in Appendix 2 of the [RGU Assessment Policy](#). It provides detail on the purpose, setting and implementation of wordage limits; lists what is included and excluded from the word count; and the penalty for exceeding the word count.

### What's included in the word count?

The table below lists the constituent parts which are included and excluded from the word limit of a Coursework; more detail can be found in the full Assessment Word Limit Statement. Images will not be allowed as a mechanism to circumvent the word count.

Excluded	Included
Cover or Title Page	Main Text e.g. Introduction, Literature Review, Methodology, Results, Discussion, Analysis, Conclusions, and Recommendations
Executive Summary (Reports) or Abstract	Headings and subheadings
Contents Page	In-text citations
List of Abbreviations and/or List of Acronyms	Footnotes (relating to in-text footnote numbers)
List of Tables and/or List of Figures	Quotes and quotations written within “...”
Tables – mainly numeric content	Tables – mainly text content
Figures	
Reference List and/or Bibliography	
Appendices	
Glossary	

### What are the penalties?

The grade for the submission will be reduced to the next lowest grade if:

- The word count of submitted work is above the specified word limit by more than 10%.
- The submission contains an excessive use of text within Tables or Footnotes.

## What else is important to my assessment?

### What is plagiarism?

Plagiarism is “the practice of presenting the thoughts, writings or other output of another or others as original, without acknowledgement of their source(s) at the point of their use in the student’s work. All materials including text, data, diagrams or other illustrations used to support a piece of work, whether from a printed publication or from electronic media, should be appropriately identified and referenced and should not normally be copied directly unless as an acknowledged quotation. Text, opinions or ideas translated into the words of the individual student should in all cases acknowledge the original source” ([RGU 2022](#)).

### What is collusion?

“Collusion is defined as two or more people working together with the intention of deceiving another. Within the academic environment this can occur when students work with others on an assignment, or part of an assignment, that is intended to be completed separately” ([RGU 2022](#)).

For further information please see [Academic Integrity](#).

### What if I'm unable to submit?

- The University operates a [Fit to Sit Policy](#) which means that if you undertake an assessment then you are declaring yourself well enough to do so.
- If you require an extension, you should complete and submit a [Coursework Extension Form](#). This form is available on the RGU [Student and Applicant Forms](#) page.
- Further support is available from your Course Leader.

### What additional support is available?

- [RGU Study Skills](#) provide advice and guidance on academic writing, study skills, maths and statistics and basic IT.
- [RGU Library guidance on referencing and citing](#).
- [The Inclusion Centre: Disability & Dyslexia](#).
- Your Module Coordinator, Course Leader and designated Personal Tutor can also provide support.

### What are the University rules on assessment?

The University Regulation '[A4: Assessment and Recommendations of Assessment Boards](#)' sets out important information about assessment and how it is conducted across the University.