Cardiff Metropolitan University | Prifysgol Metropolitan Caerdydd

School of Technologies

Assessment Brief

Module Code	Module Title
CIS6003	Advanced Programming
Academic Year	**Semester**
2025	1

Module Leader email

priyanga@icbtcampus.edu.lk

Content

Assessment Details

Assessment title	Abr.	Weighting
Online room reservation System	WRIT1	100%

Pass marks are 40% for undergraduate work and 50% for postgraduate work unless stated otherwise.

Task/assessment brief:

Scenario

Ocean View Resort is a popular beachside hotel in Galle, serving hundreds of guests each month. Currently, room reservations and guest records are managed manually, leading to booking conflicts and delays. The management now requires a **computerized system** to handle room reservations efficiently. Each guest will be assigned a **unique reservation number**. New guests must be registered in the system, which should collect details such as reservation number, guest name, address, contact number, room type, check-in date, and check-out date.

The program should provide the following functionalities:

1. **User Authentication (Login)**
   - Require a username and password for secure system access.
2. **Add New Reservation**
   - Store guest and booking details, including reservation number, guest name, address, contact number, room type, check-in, and check-out dates.
3. **Display Reservation Details**
   - Retrieve and display complete booking information for a specific reservation.
4. **Calculate and Print Bill**
   - Compute the total stay cost based on the number of nights and room rates.
5. **Help Section**
   - Provide guidelines on how to use the reservation system for new staff members.
6. **Exit System**
   - Allow users to safely close the application.

(Additional functionalities can be included as needed.)

(More functionalities can be included)

Create an error free, effective one with user friendly interfaces, appropriate messages, outputs and menu driven application using Java programming language. Students can use appropriate data structures and text files to store information.

Students are free to make necessary assumptions on system design & granting access permissions other than those mentioned within the scenario, but all suggestions must be well explained with valid reasons.

Guidelines for the report format

Paper A4 | Margins 1.5” left, 1” right, top and bottom

Page numbers – bottom, right | Line spacing 1.5

Font size

Headings 14pt, Bold | Normal 12pt

Font face- Times New Roman

Use Harvard reference to acknowledge all the external sources you use properly.

Tasks:

Tasks A:

Provide the UML diagrams for the given problem with clear explanations of the design decisions. Derive detailed Use Case diagram, Class diagram and sequence diagram. Whenever necessary document the relevant assumptions you made. **(LO I) (20 marks)**

Tasks B:

Develop an interactive System with set of interfaces to get the necessary user inputs. Make sure to implement proper validation mechanisms in order to restrict invalid entries to the system. Come up with a suitable set of reports, which you think add more value to your system

i. Your program must be a distributed application with web services

ii. Appropriate design patterns must be implemented in your system

iii. Your program should make use of a proper database to store information

(LO II) (40 marks)

Tasks C:

Document the test plan and explain how you used test-driven development in this scenario and do a test automation to achieve that. This includes test rationale, test plan, test data and proper application of the test plan **(LO II) (20 marks)**

Tasks D:

Create your own Git/ GitHub repository which is public to access and upload /deploy the changes of the software project you have developed in it. Share the report link within the documentation. Update it with several versions where modifications are applied each day, that you have applied the new features into which were initially uploaded. Version control techniques you have used throughout the development should be highlighted and documented properly. Demonstrate workflows deployed with the Git repository. **(LO III)(20 marks)**

Word count (or equivalent):	4000

This a reflection of the effort required for the assessment. Word counts will normally include source code, any text, tables, calculations, figures, subtitles and citations. Reference lists and contents of appendices are excluded from the word count. Contents of appendices are not usually considered when determining your final assessment grade.

Academic or technical terms explained:

Student advanced theoretical and with industry standards/practical knowledge of program and software analysis, design, and construction, contextualised by the use of appropriate tools, methodologies and techniques to create useful and usable software in industry.

Understand -
Involves grasping the meaning of information, ideas, or concepts. It goes beyond simply memorizing facts and requires interpreting, explaining, and summarizing the information in your own words.
Design –
Creating something new, whether it's a plan, a process, a product, or a solution to a problem. It requires using your understanding to generate something original and functional.
Develop –
Creating: This refers to taking an existing design and building upon it, refining it, or bringing it to life. This involves a high degree of originality and innovation.
Applying: This signifies using existing knowledge or skills to create something new, but not necessarily something completely original. It involves adapting and implementing existing concepts in a new context.
Evaluate –
This involves making judgments about the value, worth, or effectiveness of something. It requires analyzing information, identifying strengths and weaknesses, and comparing it to established criteria or standards.

Submission Details

Submission Deadline:	This will be provided on the Moodle submission point.	**Estimated Feedback Return Date**	This will normally be 20 working days after initial submission.
Submission Time:	By 2.00pm on the deadline day.		

Moodle/Turnitin: Any assessments submitted after the deadline will not be marked and will be recorded as a non-attempt unless you have had an extension request agreed or have approved mitigating circumstances. See the School Moodle pages for more information on extensions and mitigating circumstances.

File Format: The assessment must be submitted as a pdf document (save the document as a pdf in your software) and submit through the Turnitin submission point in Moodle.

Your assessment should be titled with your:

	student ID number, module code and assessment ID, e.g. st12345678 CSE5013 WRIT1
Feedback	Feedback for the assessment will be provided electronically via Moodle. Feedback will be provided with comments on your strengths and the areas which you can improve. View the guidance on how to access your feedback. All marks are provisional and are subject to quality assurance processes and confirmation at the programme Examination Board.

Assessment Criteria

Learning outcomes assessed

- Demonstrate fluency in contemporary programming languages, development tools and environments.
- Evaluate and demonstrate the theory and concepts of contemporary/industry-standard programming and design in the software development life cycle.
- Demonstrate awareness of industry standards of professional and ethical software development, software carpentry and codemanship.

Other skills/attributes developed

This includes elements of the Cardiff Met EDGE (Ethical, Digital, Global and Entrepreneurial skills) and other attributes developed in students through the completion of the module and assessment. These will also be highlighted in the module guidance, which should be read by all students completing the module. Assessments are not just a way of auditing student knowledge. They are a process which provides additional learning and development through the preparation for and completion of the assessment.

ETHICAL	Understanding the importance of protecting user data within java programming interactions. This includes adhering to data privacy regulations and implementing secure coding practices.
DIGITAL	Deconstructing complex problems into smaller, manageable application components and designing solutions with clear interfaces. Understanding how cloud platforms can execute and manage services, ensuring scalability and reliability.
GLOBAL	Being aware of international regulations and standards that might impact Softwere development and data handling. Embracing diverse perspectives and being flexible in adapting service design to meet the needs of global contexts.
ENTREPRENEURIAL	Identifying opportunities to create new businesses or to enhance the existing process using Software oriented application development.

Marking/Assessment Criteria

Task	Poor < 40	Satisfactory 40-49	Average 50-59	Good 60-69	Excellent 70 -100
System design with UML diagrams **(20 marks)**	No design diagrams or documentation are provided, making it difficult to understand the application. The application lacks core functionalities or is incomplete, failing to demonstrate the key services required for the case study	- Proper Use of Object-Oriented Design Methodology - Use Case Diagram - Identification of correct use cases - Identification of correct actors and associations - Sequence Diagram - Implementing an identified set of use cases (approximately 3) as sequence diagrams - Class Diagram - Identification of associated methods with correct signatures and attributes in each class - Correct identification of relationships	Use of private and public access modifiers, visible in the class diagram Use of <<include>> and <<extend>> stereotypes in the use case diagram Acceptable level of use of lifelines, messages, and objects in the proposed sequence diagrams Acceptable use of UML notations with some minor mistakes	- Clear identification of private and public access modifiers, visible in the class diagram - Accurate use of <<include>> and <<extend>> stereotypes in the use case diagram - Appropriate use of lifelines, messages, and objects in the proposed sequence diagrams - Correct use of UML notations with minor mistakes - Evaluation: - The student has provided a basic description of the design and offered reasonable justification - Effective judgments have been made	- Highly detailed diagrams - Clear use of object-oriented concepts - Supported by relevant assumptions - Multiplicity, navigability, aggregation, and composition are visible in class diagrams - Excellent use of UML notation **Evaluation: ** - Good justification of the design - Assessment of the validity of results - Use of critical reflection to evaluate the work, supported by valid explanations - Design fluency - Evidence of critical analysis from different perspectives, covering how use case, class, and sequence diagrams support the design
Design patterns and Architecture development **(40 marks)**	- No mention of design patterns - Code does not reflect the use of design patterns - Code does not reflect a tiered architecture	- Identification of various types of design patterns with a basic description of each pattern - Application of some design patterns in system development - Critical evaluation of the impact of design patterns - Basic data management system features	- The student has an acceptable level of understanding of design patterns - Application of design patterns in system development is evident in the document - The student demonstrates an acceptable level of critical evaluation of the impact of design patterns	: - Identification of different types of design patterns with descriptions and evaluations for each pattern is visible - Application of design patterns in system development is clearly evidenced in the document - The student demonstrates an acceptable level of critical evaluation of the impact of design patterns	- Identification of different types of design patterns, with descriptions and evaluations for each pattern, is visible - Application of the most suitable design patterns for system development, with clear evidence in the document - The student provides a clear explanation and critical evaluation of the impact of design patterns - More sophisticated UI

		- Use of a database (simple design) - Simple web user interface	- More sophisticated data representation (e.g., multiple classes at the business logic level) - Separate UI windows for entering results and viewing overall scores	- Good attempt to follow a three-tier architecture - More sophisticated database design and queries - More advanced data representation (e.g., multiple classes at the business logic level) - Separate UI windows for entering results and viewing overall scores	- Complex functionality (e.g., email alerts, SMS notifications, innovative features) - Use of a 3-tier architecture - Appropriate use of advanced database features (e.g., stored procedures, functions, triggers to implement business rules) - Proposed reports to facilitate decision-making - Effective use of sessions/cookies
Testing **(20 marks)**	Poor testing Document. Poor level of test pan and test cases	Provide a concise rationale for the approach adopted and discuss how test-driven development will be used. - Devise your test data. - Derive test data for the system.	Derive accepted level of test data for the system. Produce and apply a test plan Create test classes for your system in Accepted level You are to carry out relevant tests and provide documentation detailing the tests used to verify your system.	- Provide a concise rationale for the approach adopted - Discuss how test-driven development will be used - Devise test data - Derive test data for the system - Produce and apply a test plan - Create test classes for your system - Carry out relevant tests and provide documentation detailing the tests used to verify your system.	- Provide a concise rationale for the approach adopted and discuss how test-driven development will be used - Devise and derive test data for the system - Produce and apply a test plan - Create test classes for your system - Carry out relevant tests and provide documentation detailing the tests used to verify your system - Demonstrate that the code passes all tests (use screen-grabbing software and insert images into your submission) - Use of test automation - Evaluate the overall success or failure, including lessons learned - Traceability showing how each requirement is met by the design .
Documentation Work with GitHub **(20 marks)**	Errors in the documentation Poor/no Git version control, deployment, or workflows used demonstrated	Basic level of the documentation Git Repo/GitHub is used for creating a repository only and uploading the initial version only	- Accepted level of documentation standard with basic explanations - A Git repository was created, the initial version of the project was uploaded, and several versions were updated and deployed with changes. However, no	- Good standard of documentation with basic explanations - A Git repository was created, the initial version of the project was uploaded, and several versions were updated and deployed with changes. However, no workflow or version control	: - Professional standard of documentation with screenshots and clear explanations - Git repository creation, accessibility restrictions, versioning, and version control techniques demonstrated - Workflow (CI/CD) demonstrated, along

			workflow or version control techniques were demonstrated.	techniques were demonstrated.	with the deployment of changes - The latest updated version is deployed and demonstrated in the documentation

Further Information

Who can answer questions about my assessment?

Questions about the assessment should be directed to the staff member who has set the task/assessment brief. This will usually be the Module Leader. They will be happy to answer any queries you have.

Staff members can often provide feedback on an assignment plan but cannot review any drafts of your work prior to submission. The only exception to this rule is for Dissertation Supervisors to provide feedback on a draft of your dissertation.

Referencing and independent learning

Please ensure you reference a range of credible sources, with due attention to the academic literature in the area. The time spent on research and reading from good quality sources will be reflected in the quality of your submitted work.

Remember that what you get out of university depends on what you put in. Your teaching sessions typically represent between 10% and 30% of the time you are expected to study for your degree. A 20-credit module represents 200 hours of study time. The rest of your time should be taken up by self-directed study.

Unless stated otherwise you must use the **HARVARD** referencing system. Further guidance on referencing can be found in the Study Smart area on Moodle and at www.citethemrightonline.com (use your university login details to access the site). Correct referencing is an easy way to improve your marks and essential in achieving higher grades on most assessments.

Technical submission problems

It is strongly advised that you submit your work at least 24 hours before the deadline to allow time to resolve any last minute problems you might have. If you are having issues with IT or Turnitin you should contact the IT Helpdesk on (+44) 2920 417000. You may require evidence of the Helpdesk call if you are trying to demonstrate that a fault with Moodle or Turnitin was the cause of a late submission.

Extensions and mitigating circumstances

Short extensions on assessment deadlines can be requested in specific circumstances. If you are encountering particular hardship which has been affecting your studies, then you may be able to apply for mitigating circumstances. This can give the teachers on your programme more scope to adapt the assessment requirements to support your needs. Extensions and mitigating circumstances policies and procedures are regularly updated. You should refer to your degree programme or school Moodle pages for information on extensions and mitigating circumstances.

Unfair academic practice

Cardiff Met takes issues of unfair practice **extremely seriously.** The University has procedures and penalties for dealing with unfair academic practice. These are explained in full in the University's Unfair Practice regulations and procedures under Volume 1, Section 8 of the Academic Handbook. The Module Leader reserves the right to interview students regarding any aspect of their work submitted for assessment.

Types of Unfair Practice, include:

Plagiarism, which can be defined as using without acknowledgement another person's words or ideas and submitting them for assessment as though it were one's own work, for instance by copying, translating from one language to another or unacknowledged paraphrasing. Further examples include:

- Use of any quotation(s) from the published or unpublished work of other persons, whether published in textbooks, articles, the Web, or in any other format, where quotations have not been clearly identified as such by being placed in quotation marks and acknowledged.
- Use of another person's words or ideas that have been slightly changed or paraphrased to make it look different from the original.
- Summarising another person's ideas, judgments, diagrams, figures, or computer programmes without reference to that person in the text and the source in a bibliography/reference list.
- Use of assessment writing services, essay banks and/or any other similar agencies (NB. Students are commonly being blackmailed after using essay mills).
- Use of unacknowledged material downloaded from the Internet.
- Re-use of one's own material except as authorised by your degree programme.

Collusion, which can be defined as when work that that has been undertaken with others is submitted and passed off as solely the work of one person. Modules will clearly identify where joint preparation and joint submission are permitted, in all other cases they are not.

Fabrication of data, making false claims to have carried out experiments, observations, interviews or other forms of data collection and analysis, or acting dishonestly in any other way.

How is my work graded?

Assessment grading is subject to thorough quality control processes. You can view a summary of these processes on the Assessment Explained Infographic.

Grading of work at each level of Cardiff Met degree courses is benchmarked against a set of general requirements set out in Volume 1, Section 4.3 of our Academic Handbook. A simplified version of these Grade Band Descriptors (GBDs) with short videos explaining some of the academic terminology used can be accessed via the Facilitation of Learning resource page.

We would strongly recommend looking at the Study Smart area of Moodle to find out more about assessments and key academic skills which can have a significant impact on your grades. Always check your work thoroughly before submission.

Cardiff Met

Met Caerdydd