CE/CZ2002 Assignment Grading Guide
(Hotel Reservation and Payment System (HRPS)

ASSESSMENT WEIGHTAGE

UML Class Diagram [30Marks]
Show mainly the Entity classes, the essential Control and Boundary classes, and enumeration type (if there is).
Clarity, Correctness and Completeness of details and relationship.

Design Consideration [15 Marks]
Usage of OO concepts and principle - correctness and appropriateness

Implementation Code [35Marks]
Diagram to Code correctness, readability, Javanaming convention, exception handling, completeness of Java Doc and overall quality.
A Java API HTML documentation of ALL your defined classes using Javadoc must be submitted. The use of javadoc feature is documented in Appendix D.

Demonstration and report [20 Marks]
Coverage of application essentials and functionalities, user friendliness, demo flow, innovation.
Report structure and reflection

THE REPORT

Your report will include the following :

- A detailed UML Class Diagram for the application (exported as an image) show clearly the class relationship, notation notes to explain, if necessary

- A write-up on your design considerations and use of OO concepts.
  Propose 2 new features as further enhancements and write a 2-3 lines description how each feature will be used.
  Explain how your current design can cater to these 2 features- using the design principles (reusability, extensibility, SOLID, etc)

- Reflection: The difficulties encountered and the way to conquer, the knowledge learnt from this course, further improvement suggestion.

Class Diagram (30 marks)

Notes

- See the attached suggested class diagram (either versions or combination)
- These are **Entity** Classes for storing data – *usually nouns like Room, Guest, Car, etc*.
- Teams should have about the same classes.
- They may include Control and Boundary classes, eg HotelMgr, PaymentController, xxxUI, xxxForm)
- If you know about ER Diagram (Entity Relationship) in DB, you should able to relate to the Class relationships.

Assessment Guideline :

if the class diagram is too messy and hard to read, deduct 2 marks

- **25-30marks**
  - List of classes SIMILAR suggested (+ 3 more is fine), with detail listing of attributes and methods, appropriate relations and clear
- **19-24marks**
  - List of classes LESS THAN suggested, with detail listing of attributes and methods, appropriate relations and clear
- **12-18marks**
  - List of classes LESS THAN suggested, with MINIMUM of attributes and methods, appropriate relations and clear
- **7-11 marks**
  - List of classes LESS THAN suggested, with MINIMUM of attributes and methods, SPIDER WEB –like (a class relate to a lot of classes) class relationships – using Dependency is fine (the dotted slim arrow --------->)
  - OR overused of <<Interface>> { more than 4)
- **3-6 marks**
  - Only Boundary or Control classes
- **0-2 marks**
  - No submission or irrelevant classes

Remarks :

(1) Main assessment is the relationship between Entity classes. Boundary and Control classes can be considered when they justified the usage in Design Consideration.

(2) **Likely Entity classes (or similar names) :**
Reservation, Room, Guest, RoomService(Order), MenuItem.

(3) **Only a few Interfaces but NOT majority**
DBInterface, PaymentInterface

(4) **Likely Control classes (*should not be more than 4 without strong justification)***
HotelApp (main)**, PaymentXXX*, RoomXXX

Implementation (35 marks)

Notes

Deduct 7 marks if JavaDoc not available/submitted. *(include as remarks in assessment sheet)*

Assessment Guidelines

a) **Able to compile (0-5 marks)**

b) correctness of the program. (0-15 marks)

- Sample 2 big Entity classes and check the codes implementation in the .java files (or verify using the JavaDoc) is as depicted in Class Diagram.
- See the ppt slide on mapping diagram to code.

c) presentation of the Java codes (whether the program is **properly indented**); (0-4 marks)

d) documentation of codes (whether the program is well commented to aid understanding) – including JavaDoc. Check the quality details of the Java Doc html. (0-8 marks)

e) **naming convention** (camel style) - ***Class to be Uppercase*** for first letter eg Student, CourseInfo
method to be Lowercase for first letter, eg printResult (0-3 marks)

Design Consideration (15 marks)

Notes

- ***Purpose is to demonstrate their design ideas, concepts and approach taken***
- ***Everything must be written in the context of this case studies and not just quote general 'principles' without tying it to classes used in the assignment context.***
- ***How they design their text files to store the data is irrevelant BUT how the objects are used.***
- ***The consideration of reusability, maintainability and extensibility***
- ***Correct OO concepts need to be shown**.***
- ***Proposed 2 relevant feature/functions and explain it reasonably well how it can be implemented to MINIMISE impacts***

Assessment Guidelines

- Well written, have considered a number of design issues and applied design principles and OO concepts

  10-15 marks

- Considerations make to most design issues with design principles and OO concepts applied.
  - *but some ideas/concepts do not applied /used right*

  6-9 marks

- \- Quote text or principles with no mapping to **assignment context**
  - too brief to read ideas

  0-5 marks

**Remarks

It is **not necessary** to demonstrate the **use of inheritance and polymorphism** (IP)and they will **not be penalized**. IF groups used and applied (IP) in the appropriate and logical context of the assignment, better marks can be given.
The handling of the reading and writing to/from file (FileIO) are a common way approach to applying inheritance and polymorphism – eg with a DBInterface (or Data Access Object, DAO) such that in future, it can be extended to read from DB, XML or json.

Demonstration and report [20 Marks]

Assessment Guidelines

a) Coverage of application essentials and functionalities - demo stated functionality in the assignment doc

b) user friendliness - show that application is designed and used in minimum data entry (eg using selection).
c) demo flow - easy to follow and understand what is demo [from an audience perspectives trying to understand the demo]
d) clarity of narration, video display and audio-video sync
e) *Include showing of negative cases/error checking*
f) *Strong demonstration of learning points and insights of good algorithm design and implementation practices, based on experience gained from doing the assignment.*

a) to f)	All Satisfactory	**15-18 marks**	
	ONE item Not Satisfactory only	**12-14 marks**	
	Each item	3 marks	
	Not satisfactory at all	**1-2marks**	
g)	**Innovation**	**add max of 2**	

Deduct 2 marks if the video duration exceeded 15 minutes