ASSIGNMENT

CE/CZ2002: Object-Oriented Design & Programming

Building an OO Application

2015/2016 SEMESTER 2

SCHOOL OF COMPUTER ENGINEERING
NANYANG TECHNOLOGICAL UNIVERSITY

1. OBJECTIVE

The main objective of this assignment is

- to apply the Object-Oriented (OO) concepts you have learnt in the course,
- to model, design and develop an OO application.
- to gain familiarity with using Java as an object oriented programming language.
- to work collaboratively as a group to achieve a common goal.

2. LABORATORY

Computing Lab II (Location: N4-B2b-04).

3. EQUIPMENT

Hardware:	PC (or Laptop)
Software:	Your prefered Java IDE or simply notepad and Java Development ToolKits (JDK)

4. THE ASSIGNMENT

The assignment for your group will be to design and develop a:

Hotel Reservation and Payment System (HRPS).

HRPS is an application to computerize the processes of making hotel reservation, recording of orders and displaying of records. It is solely used by the hotel staff.

The following are information about the application:

About Guest

a) Guests can be added with details like name, credit card details, address, country, gender, identity, nationality, contact when check in.
b) Identity could be driving license or passport.

About Room

a) Rooms can be categorized according to its type: Single, Double, Deluxe, VIP Suite, etc. (you may refer to hotel website for the types). Each type is at different rate.
b) Rooms have details like room number, bed type (single/double/master, WIFI enabled, facing (with view), smoking/non-smoking, status, etc.
c) Room availability/details can be checked by entering the room number (id)/guest name.
d) Room availability status can be Vacant, Occupied, Reserved, Under Maintenance.

About Reservation

a) When a room is reserved, it will have a corresponding reservation details.
b) Reservations can be added with a reservation code/number, the associated guest details, room and billing information (eg credit card), date of check in/out, number of adults/children, reservation status.
c) An acknowledgement receipts with the essential reservation details should be provided (displayed) when a reservation is made.
d) Reservations can be in different status: confirmed, in waitlist, checked-in, expired, etc. Note that payment is not necessary at the time of reservation.
e) A reservation will be expired if no one checks in by a specific time point (example 1 hour after scheduled check-in time), and the room should be made available again.
f) When guests check-in, the reservation and room/s should be updated and reflect the correct status.

About Room Service

a) Hotel staff can order room service meals on guest's behalf upon his/her request.
b) List of menu items selection will be displayed for selection.
c) Each menu item will have a name, a description of how it is prepared and price.
d) When ordered, there will be a room service order created with a date/time, remarks (eg, less oil, less salt) and the status (confirmed, preparing, delivered).
e) The order status will be updated accordingly.

About Payment

a) When a guest check-out, the total bill will be presented for payment. After payment, the room/s will become available.
b) Total bill include room charges, tax, room services. Room charges can be different for weekends and weekdays. In addition, discount could be provided for promotion.
c) Payment can be made in cash, credit card with details such as billing address, etc.
d) Room occupancy report can be generated based on the percentage of occupied rooms in a particular day.

Functional Requirements:

a. Create/Update/Search guests detail (*Search by name using keyword/s*)
b. Create/Update/Remove/Print reservation
c. Create/Update rooms details (include setting status like 'Under Maintenance",
d. Entering room service orders - list menu items for selection
e. Create/Update/Remove room service menu items.
f. Check room availability
g. Room Check-in (for walk-in or reservation)
h. Room Check-out and print bill invoice (with breakdowns on days of stay, room service order items and its total, tax and total amount)
i. Print Room Status statistic report by
    a. room type occupancy rate (room status = vacant) - *(room type, number and list the room number)*.
    Eg, [for illustration only]
    Single : Number : 10 out of 20
    Rooms : 02-03, 03-04, 03-05,…….
    Double : Number : 5 out of 10
    Rooms : 02-04, 05-04, 05-05,…….
    …….

    b. room status *(status type and list the room number)*
    Eg, [for illustration only]
    Vacant :
    Rooms : 02-03, 03-04, 03-05,…….
    Occupied :
    Rooms : 02-04, 05-04, 05-05,…….
    …….

The application is to be developed as a **Console-based application (non-Graphical UI).** Data should be stored in flat file format, either in text or binary. ***No database application (eg MySQL, MS Access, etc) is to be used.***
[Refer to your eLearning Topics in NTULearn on FileI/O. Samples are provided for the reading/writing of text or binary (Serializable) file]

You will create your own test cases and data to test your application thoroughly. However, you should also create test cases to test for cases* *of full occupancy of certain room type, releasing of room/s upon payment and removing of reservation/s upon 'period expiry', presenting room occupancy with rooms under maintenance, generating of bill invoice with room service charges. Refer to* ***Appendix A* *for reference.*

Assumptions :

(1) The currency will be in Singapore Dollar (SGD).
(2) There is no need to interface with external system, eg Payment, printer, etc.
(3) Payment is always successful.
(4) No need to log in - features
(5) No graphical interface for selection of rooms.
(6) The room service bill is entered as an amount and time stamp. There is no need to specify the items consumed.
(7) The hotel has 48 rooms (of various type) from 02 – 07 level and the room number format is <2 digit floor level><running number from 01>, eg 0702.

5. THE REPORT

Your report will include the following :

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

b) A detailed UML **Sequence** Diagram (exported as an image)
- show the flow of the **"Check-out and print bill invoice"** function.
- *The diagram should show clearly all participating objects involved with sufficient detailed flow and relevant interaction fragments.*

c) 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)

d) A duly signed **Declaration of Original Work** form (Appendix B).

e) [**Optional**] Member's work contribution and distribution breakdown.
If your group feels that marks should be given based on contribution, your group can fill up the WBS.xls(in the same folder as assignment doc) and include it in this report. All members MUST consent to the WBS contents. ***You must also email the WBS.xls to the course-coordinator with ALL members in the loop.***

6. DEMONSTRATION

Your group is to produce a **video and audio recording** to demonstrate the working of the application – presenting ALL the required functionalities of the application and the suggested test cases in Appendix A. It is advised that you planned your demonstration in a story-boarding flow to facilitate understanding of your application. *Include a group photo of your group members and introduce your members and group number at the start of video.*

In the production, you may include :

a) Explaining essential and relevant information about the application
b) Run-through and elaborate on essential part/s of your implementation/coding

- ***The video duration must not exceed 15 minutes in total.***
- ***The font size used must be large enough to be readable and viewable.***
- ***The video quality must be clear.***
- ***The demo of the application is to done in real-time and NOT pre-run display.***

7. THE DELIVERABLE

Your group submission should be in the form of **CD or DVD** and should include the following :

a. The report.
b. Video and audio recording of the demonstration.
c. All implementation codes and java documentation (javadoc).

8. ASSESSMENT WEIGHTAGE

UML Class Diagram [**25 Marks**]

- 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.

UML Sequence Diagram [**20 Marks**]

- Show only the sequence Diagram mentioned in 5(b)
- Details Clarity, Correctness and Completeness of flow and object interactions.

Design Consideration [**15 Marks**]

- Usage of OO concepts and principle - correctness and appropriateness

Implementation Code [**20 Marks**]

- 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 [**20 Marks**]

- Coverage of application essentials and functionalities, user friendliness, demo flow, innovation.
- ***Based on stated video duration above.***

9. DEADLINE

This is a **group assignment**, and one CD/DVD is to be submitted from each group. Report format guidelines is provided in the Appendix below.
The CD/DVD needs to be submitted to the **Software Projects Lab @N4-B1b-11** by **15th April 2016, 4.30pm**. Indicate your group members and lab class on the CD/DVD. *Drop your CD/DVD into the pigeon hole indicating **your lab class** and the **course code CE/CZ2002**.*

Important:
Note that **THREE (3)** marks will be deducted for the delay submission of each **calendar day**.
The lab is closed on weekends and no submission can be made but late submission penalty still applies.

10. REFERENCES & TOOLS

- UML Diagrams tool - Visual Paradigm http://www.visual-paradigm.com/
- http://www.visual-paradigm.com/support/documents/vpuserguide/94/2576/7190_drawingclass.html
- NTULearn Cx2002 main course site content
- NTULearn Cx2002 course site content on "File Input/Output"
- Object Serialization tutorial http://www.javabeginner.com/uncategorized/java-serialization
- Windows Media Encoder (a suggestion) http://www.microsoft.com/expression/products/EncoderPro_Overview.aspx

APPENDIX A:

Suggested Test cases

The list of test cases are guide for your testing and demo video. Depending on your design and user-friendliness of your data entries process, there may be multiple steps taken.
[Note : You should also demonstrated at least 5 cases of input error checking done in your application]

a. To create (using the functions listed in Function requirement)
   - 2 guests occupied each room type (single, standard, VIP, suite, deluxe,) with different details, eg bed type, etc.
   - 2 rooms of each room availability status Under Maintenance.
   - 3 reservations with corresponding different guest details [** set check-in date as 1 week later*]
   - At least 5 room service menu items

b. Print Room Status statistic report by :
   - room type occupancy rate [*expected : details created above*]
   - room status [*expected : details created above*]

c. Print all reservations [*expected : details created above*]

d. Search for a guest and list its details [*found and not found*]

e. Update a guest's credit card detail
f. Search for this guest and list its details
g. Update a rooms details - change room status to 'Under Maintenance'
h. Create a reservation *[* set check-in date as 1 day later]*
   - Check availability of a room
   - After confirmation, [*acknowledgement receipts is printed with details*, a room status changed reserved]

i. Print all reservations [*expected : details created above*]
j. Print Room Status statistic report by :
   - room status - reserved and [*expected : inclusive of details created/changed above*]

k. Update the price of a room service menu item.
l. Enter room service order
   - [*listing should reflect the above change in price*]
   - select 2 menu items [*including above*]

m. Remove a room service menu item.
n. Enter room service order
   - [*listing should reflect the above change in price*]
   - select 1 menu items

o. Room Check-out and print bill [*change system date to show 5 days of stay*]
p. Check availability of a room [*above room should be available*]

q. Print all reservations [*expected : *expired reservation should not be shown*]

r. Room Check-in
   - For Walk-In
   - For one of the reservation**

s. Print Room Status statistic report by :
   - room type occupancy rate [*expected : details created above*]
   - room status [*expected : details created above, ** reserved -> occupied*]

t. Print Room Status statistic report by :
   - room type occupancy rate [*expected : details created above*]
   - room status [*expected : details created above*]

APPENDIX B:

Declaration of Original Work for CE/CZ2002 Assignment

We hereby declare that the attached group assignment has been researched, undertaken, completed and submitted as a collective effort by the group members listed below.

We have honored the principles of academic integrity and have upheld Student Code of Academic Conduct in the completion of this work.

We understand that if plagiarism is found in the assignment, then lower marks or no marks will be awarded for the assessed work. In addition, disciplinary actions may be taken.

Name	Course (CE2002 or CZ2002)	Lab Group	Signature /Date

Important notes:

1. Name must **EXACTLY MATCH** the one printed on your Matriculation Card.

APPENDIX C:

Report requirement:

1. Format:

For the main content, please use Times New Roman 12 pt font size and 1.5 line spacing. You may choose to use other fonts (e.g, Courier New) for code segments. Please use the following report structure:

- Cover page: Declaration of original work (Page 10 of the assignment)
- Design Considerations .
  - Approach taken, Principles used, Assumptions made, etc
  - ***Optional*** : You can show the important code segment (e.g, a method or a few lines of code) and necessary illustrations to explain your solution.
- Detailed UML Class Diagram.
  - Further Notes, if needed
- Detailed UML Sequence Diagram of stated function.
  - Further Notes, if needed
- Testing.
  - Test Cases and Results

2. Length:

The report should be at most 11 pages from cover to cover including diagrams/Testing results/references/appendix, if there is any. If you could well present your work in fewer than 11 pages, you are encouraged to do so.

DO NOT include source code in the report but stored the source code in the CD/DVD.

APPENDIX D:

Creating Javadoc:

Detailed can be found at http://www.oracle.com/technetwork/java/javase/documentation/index-137868.html

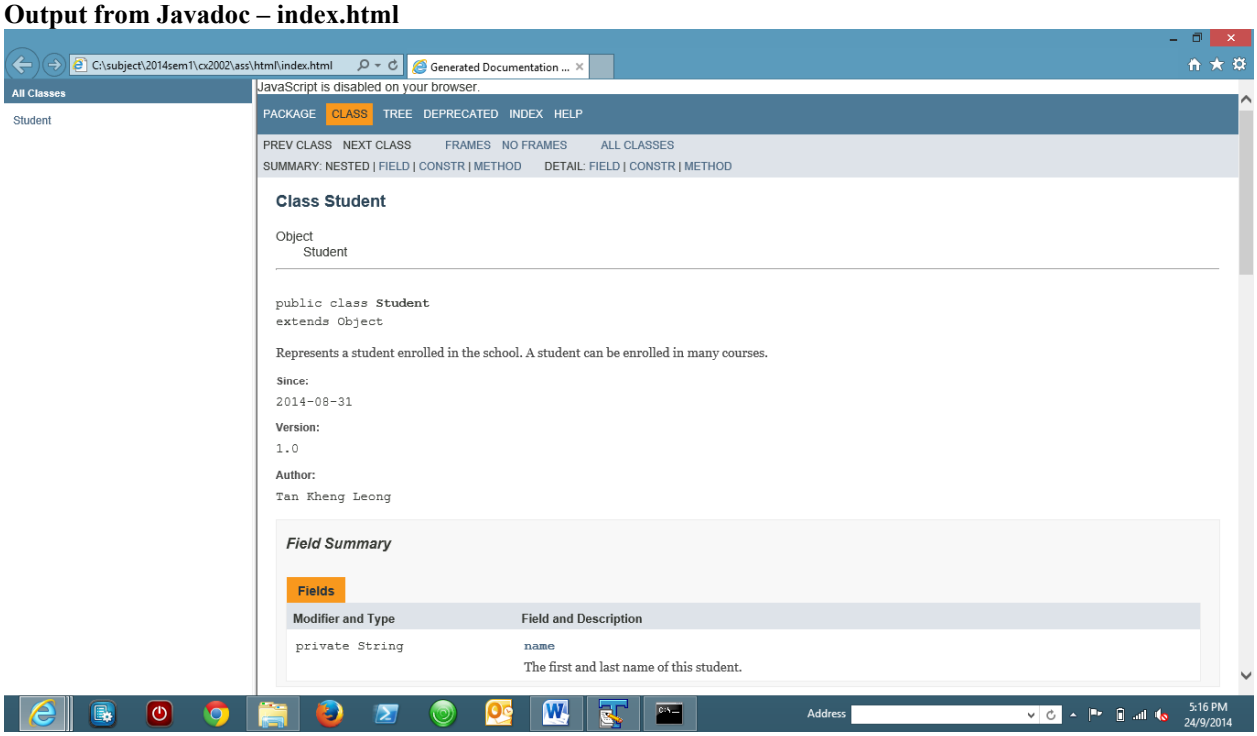
Using Javadoc in Eclipse : Youtube : http://www.youtube.com/watch?v=Hx-8BD_Osdw

Below is a short example :

```
/**
  Represents a student enrolled in the school.
  A student can be enrolled in many courses.
  @author  Tan Kheng Leong
  @version 1.0
  @since   2014-08-31
 */
public class Student {

  /**
   * The first and last name of this student.
   */
  private String name;

  /**
   * The age of this student.
   */
  private int age;

  /**
   * Creates a new Student with the given name.
   * The name should include both first and
   * last name.
   * @param name This Student's name.
   * @param age  This Student's age.
   */
  public Student(String name, int age) {
    this.name = name;
    this.age = age;
  }

  /**
   * Gets the first and last name of this Student.
   * @return this Student's name.
   */
  public String getName() {
    return name;
  }

  /**
   * Changes the name of this Student.
   * This may involve a lengthy legal process.
   * @param newName This Student's new name.
   *                Should include both first
   *                and last name.
   */
  public void setName(String newName) {
    name = newName;
  }
```

```
}
```

Output from Javadoc – index.html



<u>For those familiar with using command prompt :</u>

Steps to general API doc :

(1) Locate the installed path of JDK (java development kit)
- In Windows, it should be in C:\Program Files\Java\jdk<version>\

(2) Open command prompt

(3) Go to your src directory using cd

(4) At prompt …..src> <path to jdk>\bin\javadoc" -d ./html -author -private -noqualifier all -version <packagename1> <packagename2> <….>

Eg .

C:\subject\2014sem1\cx2002\src>"C:\Program Files (x86)\Java\jdk1.8.0_05\bin\javadoc" -d ./html -author -private -noqualifier all -version edu.ntu.sce.cx2002 edu.ntu.sce.cx2003

Statement	**Purpose**
C:\subject\2014sem1\cx2002\src>	Path to your src root
"C:\Program Files (x86)\Java\jdk1.8.0_05\bin\javadoc"	Path to your jdk javadoc.exe [using double quote if path has space in between, eg Program Files]
-d ./html	-d : specific folder to store html doc Eg ./html means current directory create a html folder to store
-author	Include @author in doc, if provided
-private	Include all methods and fields
--noqualifier all	Omitted all full package name. Eg show **String** instead of **java.lang.String**
-version	Include @version in doc, if provided
edu.ntu.sce.cx2002 edu.ntu.sce.cx2003	Different package names