



ASSIGNMENT

CE/CZ2002: Object-Oriented Design & Programming

Building an OO Application

2018/2019 SEMESTER 2

**SCHOOL OF COMPUTER SCIENCE AND 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**

Software Lab II (Location: N4-B1c-06).

3. **EQUIPMENT**

Hardware: PC (or Laptop)

Software: Your preferred 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 :

Restaurant Reservation and Point of Sale System (RRPSS).

RRPSS is an application to computerize the processes of making reservation, recording of orders and displaying of sale records. It will be solely used by the restaurant staff.

The restaurant operates in 2 sessions (AM : 11am – 3pm, PM : 6pm – 10pm).

The following are information about the application:

- a) Menu items should be categorized according to its type, eg, Main course, drinks, dessert, etc.
- b) Menu items can be added with details like name, description, price, etc.
- c) Promotional set package comes in a single package price with descriptions of the items to be served.
- d) A customer may order a set package or ala carte menu items.
- e) An order should indicate the staff who created the order.
- f) Staff information can be in the form of name, gender, employee ID and job title.
- g) Reservation is made by providing details like date, arrival time, #pax, name, contact number, etc and the table status is 'reserved'. The system should check availability and allocate a suitable table.
- h) Contact number is used to identify reservation.
- i) When a reservation is made, the table is reserved till the reservation booking is removed (eg time expired). Once a table is reserved, it cannot be booked for that particular session (AM/PM).
- j) Once an order is entered, the table status is 'occupied'*.
- k) Once an order invoice is printed, it is assumed that payment has been made and the table status is 'vacated'*.
- l) Table comes in different seating capacity, in even sizes, with minimum of 2 and maximum of 10 pax ("Persons At Table").
- m) Order invoice can be printed to list the order details (eg, table number, timestamp) and a complete breakdown of order items details with taxes details.
- n) Sale revenue report will detail the period, individual sale items (*either ala carte or promotional items*) and total revenue.

Functional Requirements:

1. Create/Update/Remove menu item
2. Create/Update/Remove promotion
3. Create order
4. View order
5. Add/Remove order item/s to/from order
6. Create reservation booking
7. Check/Remove reservation booking
8. Check table availability
9. Print bill invoice
10. Print sale revenue report by period (eg day or month)

(Note : you may re-order or re-phrase the above functionalities when displaying your application menu)

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. Refer to your eLearning Topics in NTULearn on FileI/O. Samples are provided for the reading/writing of text or binary (Serializable) file. [Learn from the fundamentals].

No database application (eg MySQL, MS Access, etc) is to be used. No JSON or XML is to be used.

You may populate your menu items with data collected from the internet.

You will create your own test cases and data to test all the functions thoroughly. However, you should also create test cases to test for cases* of full reservation, releasing of table/s upon payment, removing of reservation/s upon 'period expiry' and generating of bill invoice.

Assumptions :

- (1) Reservation can only be made in at most 1 month in advance.
- (2) Reservation will be automatically removed if the guest does not arrive within **XX** minutes (eg 30 minutes) after the stated arrival time*.
- (3) The currency will be in Singapore Dollar (SGD) and Good and Services Tax (GST) and service charge must be included in the order invoice.
- (4) There is **no** requirement for access control and there is **no need** for authentication (login/logout) in order to use the application.
- (5) There is no need to interface with external system, eg Payment, printer, etc. Payment via credit card will always be successful.
- (6) *The restaurant has 30 tables – 5 x 10-seats, 5 x 8-seats, 10 x 4-seats, 10 x 2-seats.*
- (7) Tables cannot be combined/joined (eg, join 3 x 2-seats table to form 6-seats table) and the table ID is fixed.

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)
 - Write the user case(function) flow description of the **“Print bill invoice” function**.
 - Show the flow of the **“Print bill invoice” function** in the Sequence Diagram.
 - The flow should start from the staff entering the table up to the point where the bill invoice is displayed/printed with a breakdown of each item details (refer to Appendix A as an example).
 - 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).

All group members must sign the 'Declaration of Original Work for CE/CZ2002 Assignment' and scan it to include in your report (or a separate file). **[5 marks deduction apply for non-compliance, plagiarism clauses still applies]**.

What is plagiarism?

<https://ntulearn.ntu.edu.sg/bbcswebdav/courses/AI0001-Master/m/what-is-plagiarism.htm>

What is the penalties for plagiarism?

<https://ntulearn.ntu.edu.sg/bbcswebdav/courses/AI0001-Master/m/penalties-for-plagiarism.htm>

- 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 storyboarding 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 :

- Explaining essential and relevant information about the application
 - 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 include the following:

- The report and clear image files of the diagrams.
- Video and audio recording of the demonstration.
- 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).

You are assessed on the Clarity, Correctness and Completeness of details and relationship.

UML Sequence Diagram [20 Marks]

- Write the user case(function) flow description for 5(b)
- Draw the Sequence Diagram for 5(b) aligned with the flow description.

You are assessed on the Details Clarity, Correctness and Completeness of flow and object interactions.

Design Consideration [15 Marks]

You are assessed on Usage of OO concepts and principle - correctness and appropriateness

Implementation Code [20 Marks]

- Include 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.

You are assessed on Diagram to Code correctness, readability, Java naming convention, exception handling, completeness of Java Doc and overall quality.

Demonstration [20 Marks]

- *Demonstration of the application functions based on stated video duration above.*

You are assessed on the Coverage of application essentials and functionalities, user friendliness, demo flow and innovation.

9. **DEADLINE**

This is a **group assignment**, and one CD/DVD is to be submitted from each group. Report format guidelines are provided in the Appendix C below.

1. Soft copy of the **report and video to be uploaded** to your individual CE/CZ2002 **LAB site** (eg FEP1, FSP1, etc). The link is provided on the left panel "Assignment Submission".

File name convention : <lab_grp>-grp<assignment_grp#>.<ext>

Eg, FEP2-grp3.pdf [**<ext>** can be pdf, doc, zip, mpeg, wmv or mp4]

2. **SUBMISSION DEADLINE :** **19th April 2019, 11.59pm.**

Important:

Note that THREE (3) marks will be deducted for the delay submission of each calendar day.

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:**An examples of a restaurant bill.**

<p style="text-align: center;"> TRU ***** 676 N. St. Clair Chicago, IL 60611 </p> <p style="text-align: right;">14922</p> <p> TEAM B Thu 09/09/99 9:55 PM </p> <hr/> <table> <tr><td>3 GL CLOS DU VAL</td><td>39.00</td></tr> <tr><td>3 'GL-TALLEY CHAR</td><td>39.00</td></tr> <tr><td>1 TALLEY</td><td>49.00</td></tr> <tr><td>4 PRE FIXE COLLEC</td><td>260.00</td></tr> <tr><td>1 [SIDE LOB MASH]</td><td>9.00</td></tr> <tr><td>3 COFFEE</td><td>9.00</td></tr> </table> <hr/> <p style="text-align: right;">SubTotal 405.00 Taxes... 39.50</p> <p style="text-align: right;">Total. 444.50</p> <p style="text-align: center;">***** * Thank you for dining with us! * *****</p>	3 GL CLOS DU VAL	39.00	3 'GL-TALLEY CHAR	39.00	1 TALLEY	49.00	4 PRE FIXE COLLEC	260.00	1 [SIDE LOB MASH]	9.00	3 COFFEE	9.00	<p style="text-align: center;"> Sidecar Bar & Grill 577 College Street Toronto, Ontario Tel: 416-536-7000 Check #: 39580 </p> <p> Server: Deb Table: 11 </p> <p style="text-align: right;"> Date: 12/06/2011 Time: 21:26 Client: 2 </p> <hr/> <table> <tr><td>1 Oysters</td><td>6.00</td></tr> <tr><td>1 Smelts</td><td>12.00</td></tr> <tr><td>1 Duck Terrine</td><td>14.00</td></tr> <tr><td>1 Prix Dessert</td><td>0.00</td></tr> <tr><td>1 AG Malbec</td><td>25.00</td></tr> <tr><td>1 Soup</td><td>8.00</td></tr> <tr><td>1 Tagliatelle</td><td>16.00</td></tr> <tr><td>1 D&S Brownie</td><td>9.00</td></tr> </table> <hr/> <p style="text-align: right;">SUB-TOTAL: 90.00 Hst: 11.70</p> <hr/> <p style="text-align: right;">TOTAL : 101.70</p> <hr/> <p style="text-align: center;">Join us for our \$25 Prix Fixe Sunday - Wednesday</p> <p style="text-align: center;">GST: 828874412RT0001</p>	1 Oysters	6.00	1 Smelts	12.00	1 Duck Terrine	14.00	1 Prix Dessert	0.00	1 AG Malbec	25.00	1 Soup	8.00	1 Tagliatelle	16.00	1 D&S Brownie	9.00
3 GL CLOS DU VAL	39.00																												
3 'GL-TALLEY CHAR	39.00																												
1 TALLEY	49.00																												
4 PRE FIXE COLLEC	260.00																												
1 [SIDE LOB MASH]	9.00																												
3 COFFEE	9.00																												
1 Oysters	6.00																												
1 Smelts	12.00																												
1 Duck Terrine	14.00																												
1 Prix Dessert	0.00																												
1 AG Malbec	25.00																												
1 Soup	8.00																												
1 Tagliatelle	16.00																												
1 D&S Brownie	9.00																												
<p style="text-align: center;"> Mandarin Orchard Singapore Chatterbox CoffeeHouse </p> <hr/> <p> 31/1 CHK 291 GST 2 28OCT'11 </p> <hr/> <table> <tr><td>1 Chicken Rice</td><td>27.00</td></tr> <tr><td>1 Chicken Curry</td><td>24.00</td></tr> </table> <p style="text-align: right;"> SUBTOTAL..... 51.00 10% SERVICE CHRG 5.10 56.10 7% GST 3.93 10:12 TOTAL DUE. \$60.05 </p> <p style="text-align: center;"> Tel :67374411 / Fax :67322361 333 Orchard Road S'pore 238867 </p>	1 Chicken Rice	27.00	1 Chicken Curry	24.00																									
1 Chicken Rice	27.00																												
1 Chicken Curry	24.00																												

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;
    }

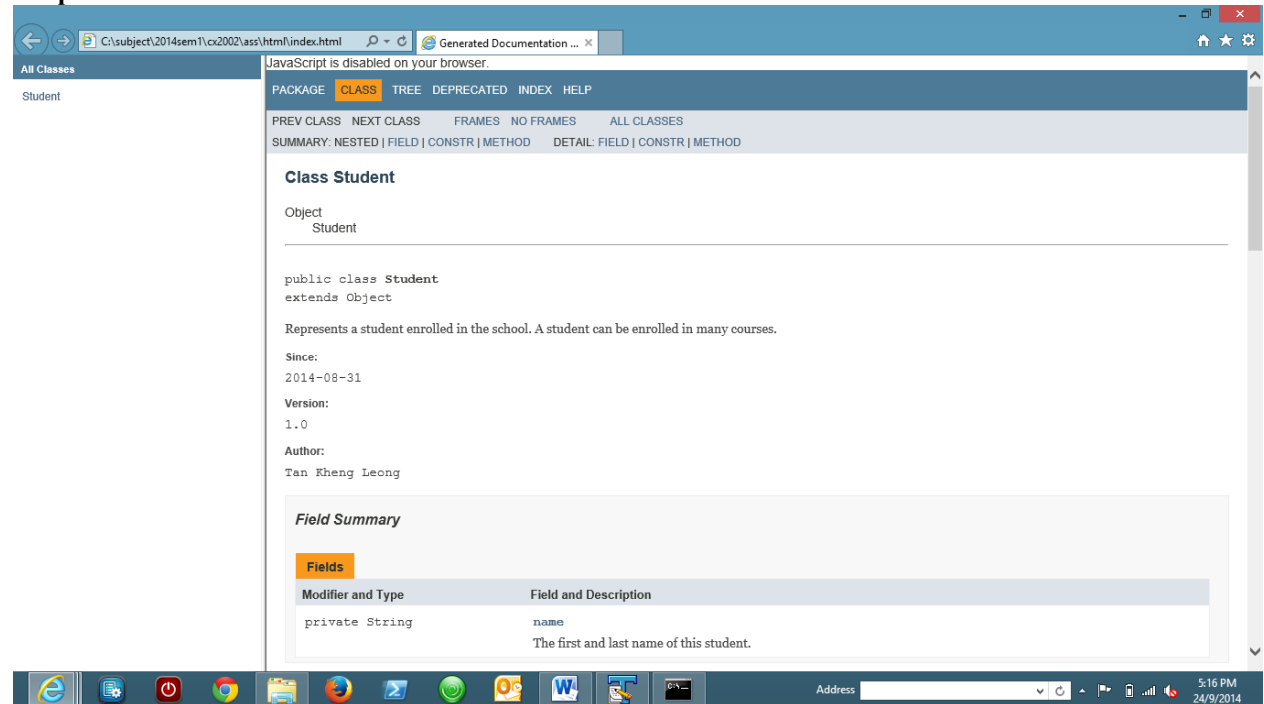
    /**
     * Gets the first and last name of this Student.
     * @return this Student's name.
     */
    public String getName() {
        return name;
    }

    /**
     * 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



For those familiar with using command prompt :

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 promptsrc> <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