ASSIGNMENT

CE/CZ2002: Object-Oriented Design & Programming

Building an OO Application

2014/2015 SEMESTER 1

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 :

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 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.
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 and ala carte menu items as well.
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, time, #pax, name, contact, etc. The system should check availability and allocate a suitable table.
h) When a reservation is made, a table is reserved till the reservation booking is removed.
i) Table comes in different seating capacity, in even sizes, with minimum of 2 and maximum of 10 pax ("Persons At Table").
j) 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.
k) Discounts can be given to customers who hold membership of the restaurant or other entities. attribute isMember
l) 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 order 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. ***No database application (eg MySQL, MS Access, etc) 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 your application thoroughly. However, you should also create test cases to test for cases***** of full reservation, releasing of table/s upon payment and removing of reservation/s upon 'period expiry'.

Assumptions :

(1) Reservation can only be made in advance. Reservation will be automatically removed **XX** minutes after the actual booking time*****.
(2) The currency will be in Singapore Dollar (SGD) and Good and Services Tax (GST) and service charge must be included in the order invoice.
(3) Once an order invoice is printed*****, it is assumed that payment has been made and the table is vacated*****.
(4) Customer with membership card will be entitled to discount.
(5) There is **no** requirement for access control and there is **no need** for authentication (login/logout) in order to use the application.
(6) There is no need to interface with external system, eg Payment, printer, etc.

5. THE REPORT

Your report will include the following :

a) A detailed UML Class Diagram for the application
   - show clearly the class relationship, notation
   - notes to explain, if necessary

b) A write-up on your design considerations and use of OO concepts.

c) A detailed UML Sequence Diagram showing the flow of the **"Check/Remove reservation booking"** function.
   - Your flow should include the case of *removing of reservation/s upon 'period expiry'*.
   - The diagram should show clearly all participating objects (minimum 3) involved with relevant interaction fragments and flow details to understand your design fully.

d) Screen captures of the testings done (those essential test cases not covered in your demo).

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

f) **[Optional]** Member's work contribution and distribution breakdown.
Example, by class implementation/s, by diagram creation, etc.

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. 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.

8. ASSESSMENT WEIGHTAGE

UML Class Diagram [**25 Marks**]

- Clarity, Correctness and Completeness

UML Sequence Diagram [**20 Marks**]

- Clarity, Correctness and Completeness

Design Consideration [**15 Marks**]

- Usage of OO concepts and principle - correctness and appropriateness
- You should explain how your design handles :
  *When a menu item is removed or its price is changed, does it affect the integrity of the past (before the change) sale invoices?

Implementation Code [**20 Marks**] ***[individual]***

- Diagram to Code correctness, readability, naming convention, exception handling and overall quality.
- Note that this an individual assessment and each member should have equal share of the class implementation.
- 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.
- In the Class Diagram, change the color of the each Class element to reflect the member assigned to implement the class, eg Green to represent StudentA, etc.
  [*In Visual Paradigm, right click on the class in the Class Diagram > Styles and Formatting > Formats….*]

Demonstration [**20 Marks**]

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

[* : essential test cases to run through in your demonstration]

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 **14th November 2014, 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**.*

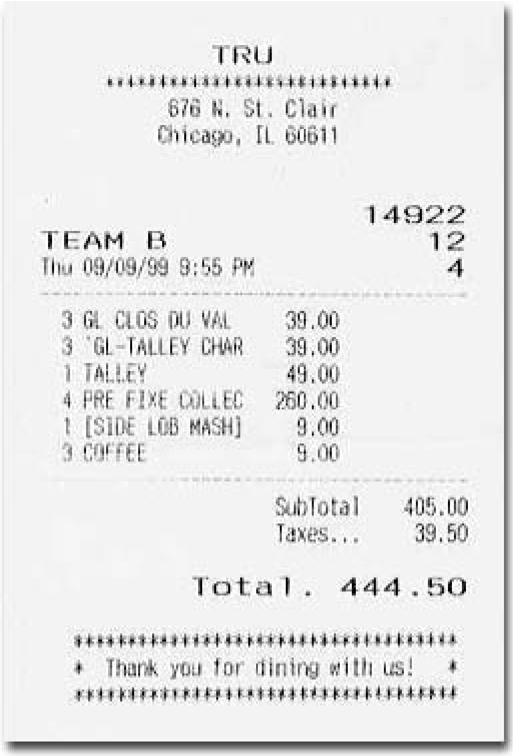
Note that **THREE (3)** marks will be deducted for the delay submission of each **calendar day**.

10. REFERENCES & TOOLS

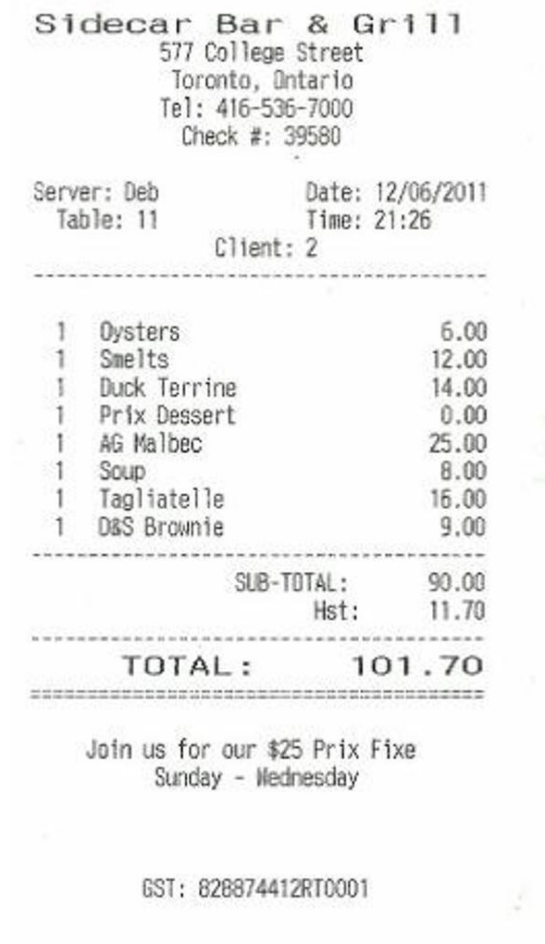
- UML Diagrams tool - Visual Paradigm http://www.visual-paradigm.com/
- Edventure Cx2002 main course site content
- Edventure Cx2002 course site content on “File Input/Output”
- Object Serialization tutorial http://www.javabeginner.com/uncategorized/java-serialization
- Windows Media Encoder (a suggestion) http://www.microsoft.com/expression/products/EncoderPro_Overview.aspx

APPENDIX A:

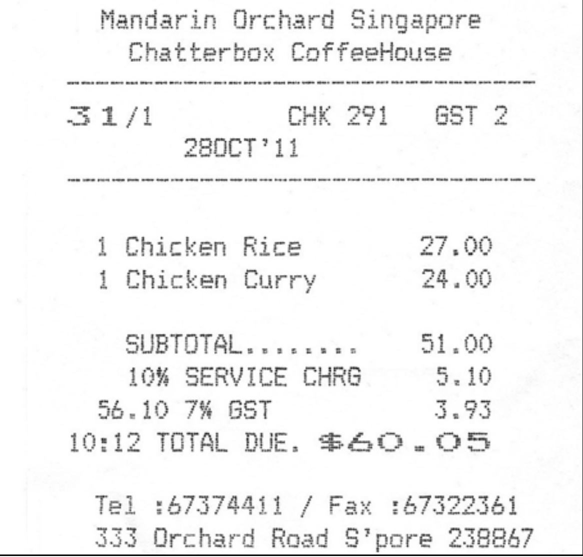
An examples of a restaurant bill.



```
              TRU
  ****************************
          676 N. St. Clair
         Chicago, IL 60611

                              14922
TEAM B                           12
Thu 09/09/99 9:55 PM              4
-----------------------------------
  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
-----------------------------------
                 SubTotal    405.00
                 Taxes...     39.50

             Total.  444.50

  **********************************
  *  Thank you for dining with us!  *
  **********************************
```



```
Sidecar Bar & Grill
          577 College Street
          Toronto, Ontario
          Tel: 416-536-7000
          Check #: 39580

Server: Deb              Date: 12/06/2011
  Table: 11              Time: 21:26
                Client: 2
------------------------------------------

  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
------------------------------------------
                   SUB-TOTAL:       90.00
                         Hst:       11.70
------------------------------------------
     TOTAL:            101.70
==========================================

      Join us for our $25 Prix Fixe
          Sunday - Wednesday


          GST: 828874412RT0001
```



```
   Mandarin Orchard Singapore
     Chatterbox CoffeeHouse
-------------------------------------
31/1          CHK 291    GST 2
         28OCT'11
-------------------------------------


  1 Chicken Rice          27.00
  1 Chicken Curry         24.00

    SUBTOTAL........      51.00
    10% SERVICE CHRG       5.10
  56.10 7% GST             3.93
10:12 TOTAL DUE. $60.05

  Tel :67374411 / Fax :67322361
  333 Orchard Road S'pore 238867
```

APPENDIX B:

Attached a scanned copy with the report with the filled details and signatures.

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/Java API doc in the report but stored the them in the CD/DVD.

APPENDIX D:

Creating Javadoc:

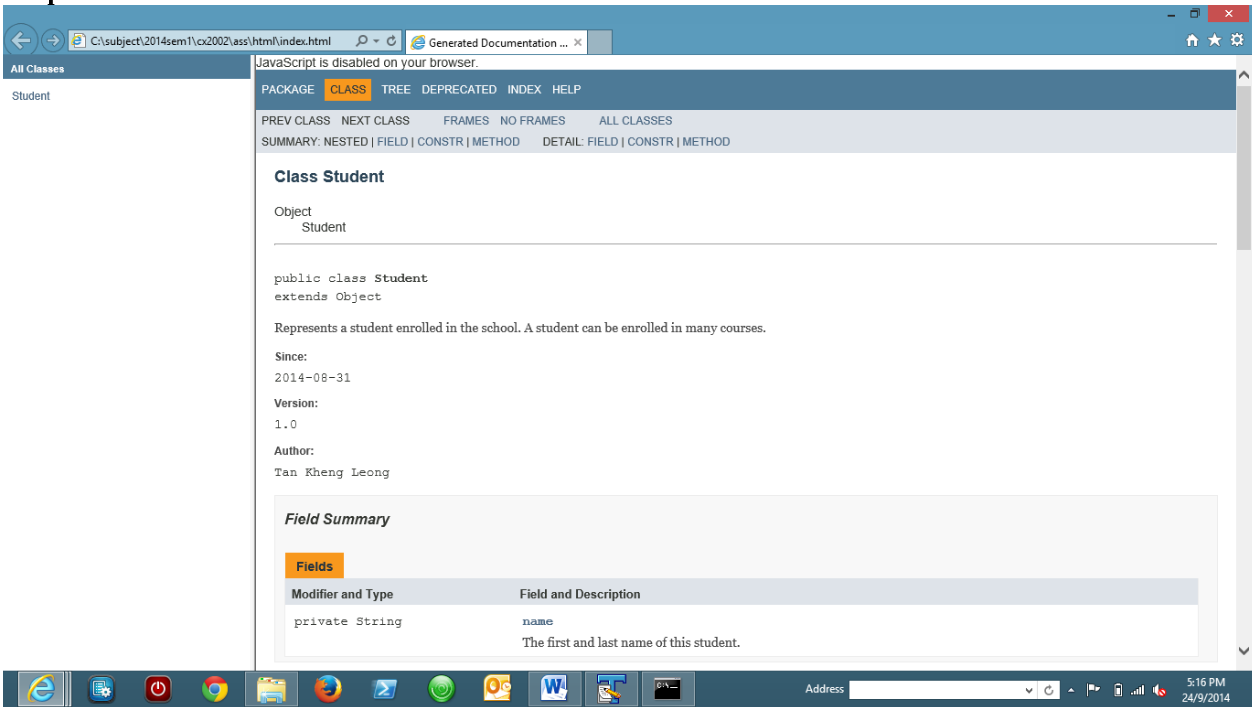
Detailed can be found at http://www.oracle.com/technetwork/java/javase/documentation/index-137868.html

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


C:\subject\2014sem1\cx2002\ass\html\index.html
Generated Documentation ...
All Classes
Student
PACKAGE CLASS TREE DEPRECATED INDEX HELP
PREV CLASS NEXT CLASS FRAMES NO FRAMES ALL CLASSES
SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD
Address
5:16 PM
24/9/2014

JavaScript is disabled on your browser.

Class Student

Object
Student

```
public class Student
extends Object
```

Represents a student enrolled in the school. A student can be enrolled in many courses.

Since:
2014-08-31

Version:
1.0

Author:
Tan Kheng Leong

Field Summary

Fields

Modifier and Type	Field and Description
private String	name The first and last name of this student.

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 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