SC/CE/CZ2002: Object-Oriented Design & Programming

ASSIGNMENT

Building an OO Application

2023/2024 SEMESTER 1

SCHOOL OF COMPUTER SCIENCE & 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 programminglanguage.
- to work collaboratively as a group to achieve a common goal.

2. LABORATORY

Assigned SCSE lab.

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:

Camp Application and Management System (CAMs).

CAMs is an application for staff and students to manage, view and register for camps within NTU. The application will act as a centralized hub for all staff and students.

User:

- All users will need to login to this hub using their user account.
  - User ID will be the NTU network user ID, that is the part before @ in email address.
  - Assume all users use the default password, which is password.
  - A user can change password in the system.
  - A user will have faculty information. E.g, EEE, SCSE.

- A student list and staff list can be initiated through files uploaded into the system at initialization. The sample student file and staff file are provided.

Staff:

- A staff will be able to create, edit and delete camps.
- A staff can toggle the visibility of the camp to be “on” or “off”. This will be reflected in the camp list that will be visible to students.
- A staff can view all camps.
- A staff can see list of camps that his/her created in a separate menu list so they can edit the camps they created.
- A staff can view and reply to enquiries from students to the camp(s) his/her has created.

- A staff can view and approve suggestions to changes to camp details from camp committee.
- A staff can generate a report of the list of students attending each camp that his/her has created. The list will include details of the camp as well as the roles of the participants. There should be filters for how the staff would want to generate the list. (attendee, camp committee, etc.) (generate in either *txt* or *csv* format).
- A staff can also generate a performance report of the camp committee members.

Camp information entered by the staff will include information like:

- Camp Name
- Dates
- Registration closing date
- User group this camp is open to own school or whole NTU
- Location
- Total Slots
- Camp Committee Slots (max 10)
- Description
- Staff in charge (automatically tied to the staff who created it)

Camp

- Each camp should have **Camp information** as above**.**
- Each camp will include a list of the students that have registered for it as camp attendees and camp committee.
    - Staff and camp committee members will be able to access this list.

Student:

- A student can only view the list of camps that are open to his/her user group (SCSE, whole NTU etc.) and if their visibility has been toggled "on".
- A student can view the remaining slots of each camp that is open to his/her.
- A student will be able to register for camps either as a camp attendee or camp committee.
- A student can submit enquiries regarding a camp.
    - Only staff and camp committees in charge of that camp can view it.
- A student can view, edit, and delete their enquiries before it is processed.
- The status of the student as a camp committee will be reflected in their profile.
- A student is only able to be in the camp committee for one camp but can attend multiple camps.
- A student is not allowed to register for multiple camps if there are clashes in the dates.
- A student only can register a camp before it is full.
- A student only can register a camp before it's registration deadline.
- A student can see the camps that his/her has already registered for and his/her roles (attendees OR camp committee)
- A student is allowed to withdraw from camps that his/her has already registered for. The remaining slot will be updated automatically. But the student is not allowed to register the same camp again.

Camp committee member

- A camp committee member can view the details of the camp that he/she has registered for.
- A camp committee member can submit suggestions for changes to camp details to

staff.

- A camp committee member is not allowed to directly edit the camp details.
- A camp committee member can view and reply to enquiries from students to the camp they oversee.
- A camp committee member can view, edit, and delete the details of his/her suggestions before being processed.
- A camp committee member can generate a report of the list of students attending each camp that they have created. The list will include details of the camp as well as the roles of the participants. There should be filters for how the camp committee member would want to generate the list. (attendee, camp committee, etc.) (generate in either *txt* or *csv* format).
- A camp committee member can get one point for each enquiry replied and each suggestion given. One extra point will be granted for each accepted suggestion.
- A camp committee member cannot quit from camp.

Assumption:

- All users can use filters to view the camp list (date, location etc.) Assume that default is by alphabetical order.
- We assume that registration of camp and camp committee is automatic as long as there is vacancy.
- We assume that the number of camp committee is counted into total slots.

The application is to be developed as a **Command Line Interface (CLI) application (non-Graphical User Interface).**

The sample data file for user list is given in excel in assignment folder. You can

- **use them directly,**
- **or copy the content to text file if you plan to read from text file,**
- **or make your own data files.**

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

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 **write-up** on your **design considerations** and use of OO concepts in your current design, extensibility and maintainability of your design.

c) Reflection: The difficulties encountered and the way to conquer, the knowledge learnt from this course, further improvement suggestion. Strong demonstration of learning points and insights of good design and implementation practices, based on experience gained from doing the assignment.

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 & Presentation (Deadline: Tuesday of week 15)**

Your group is required to present your work to your TA 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. *Please introduce your members and group number at the start of presentation, all the group members must take turn to present. The presenter should show his/her face while presenting.*

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 presentation duration must not exceed 15 minutes in total.***
- ***The font size used must be large enough to be readable and viewable.***
- ***The demo of the application is to done in real-time and NOT pre-run display.***
- ***The presentation can be conducted either through online meeting or physically.***

You will create your own test cases and data to test your application thoroughly. *Refer to* ***Appendix A* *for reference.*

7. **THE DELIVERABLE (Deadline: Sunday of Week 14)**

Your group submission should include the following:

a. The report (separate diagram file if diagram is unclear in report)
b. All implementation codes and java documentation (javadoc).
c. Other relevant files (eg data files, setup instruction, etc)

8. **ASSESSMENT WEIGHTAGE**

UML Class Diagram [**25 Marks**]

- Show mainly the Entity classes, the essential Control and Boundary classes, andenumeration type (if there is).
- Clarity, Correctness and Completeness of details and relationship.

Design Consideration [**20 Marks**]

- Usage of OO concepts and principle - correctness and appropriateness
- Explanation of design choices and how it fits the project requirements
- Coupling and cohesion of classes

Implementation Code [**30 Marks**]

- Diagram to Code correctness, readability, Java naming convention, exception handling, completeness of Java Doc and overall quality.
- A Java API HTML documentation of **ALL** your defined classes using Javadoc mustbe submitted. The use of javadoc feature is documented in Appendix D.

Demonstration and report [**25 Marks**]

- Coverage of application essentials and functionalities, user friendliness, demoflow, innovation.
- Report structure and reflection

9. SUBMISSION

This is a **group assignment**, and one submission from each group.

Report format guidelines are provided in the Appendix C below.

1. Soft copy of your deliverables to be **uploaded** to your individual SC/CE/CZ2002 **LAB site** (eg FEP1, FSP1, etc) in **NTULearn**. 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,]
2. **DEADLINE** : **Week 14 Sunday, 11.59pm.**

Important:

Note that THREE (3) marks will be deducted for the delay submission of each calendar day. Lateness is based on the date the captured in NTULearn or subsequent resubmissions (whichever is later). So check your work before submitting.

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
  [You can also try with the video recording feature for gaming in Windows 10 – press 'Windows key + G']

APPENDIX A:

Suggested Test cases

The list of sample test cases is a 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 yourapplication]

a. Login
   - Cannot login: invalid user or valid ID but wrong password
   - After login successfully, different menu list (main page) displayed for different user category, **staff**, **students,** and **camp committee**.
   - After login, the user can change password. [expected: re-login to verify the effect]
   -

b. The main page for a student should have functions after login as follows (the details should strictly follow the above description in Section 4):
   - Change password.
   - View available camps according to their faculty and visibility toggle.
   - Select the camps to register as camp attendee or committee.
   - Submit enquiries for a camp
   - View his/her own registered camps.
   - View reply to enquiry.
   - Request to withdraw from camps.

c. The main page for a staff should have functions after login as follows (the details should strictly follow the above description in Section 4):
   - Change password.
   - Create/edit/View all camps. (Only a staff can view all camps.)
   - Input choice from 'Create', 'Edit', 'View' to choose function
   - Create a camp by inputting the required data.
   - Edit: (only able to edit camps he/she has created; a staff should not be able to edit camps from other staff).
   - View list of students that have registered for the camp as attendees or committee can be viewed.
   - View Suggestions to changes to camp details from camp committee.
   - Accept/Reject suggestion
   - Generate report for each camp:
     - Camp report with the list of students attending each camp.
     - Camp committee performance report
     - Students' enquiry report.

d. The main page for a Camp committee memeber should have functions after login as follows (the details should strictly follow the above description in Section 4):
   - **All the functions that stdudent main page have**
   - Submit suggestions to staff for changes to camp details.
   - View and reply to enquiries from students from the camp he/she oversee.

- View, edit and delete the details of his/her unprocessed suggestions
- Generate report for each camp:
  - Camp report with the list of students attending each camp.
  - Students' enquiry report.

e. The main page for everyone should have functions after login as follows (the details should strictly follow the above description in Section 4):
   - If the user is logging into the system for the first time, prompt th user to change password.

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 beawarded 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.
2. Student Code of Academic Conduct includes the latest guidelines on usage of Generative AI and any other guidelines as released by NTU.

APPENDIX C:

Report requirement:

1. **Format**:

For the main content, please use Times New Roman 12 pt font size and 1.5 linespacing. 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 (Appendix B)
- 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 toexplain your solution.
- Detailed UML Class Diagram.
  - Further Notes, if needed
- Testing.
  - Test Cases and Results
- Reflection.
  - The difficulties encountered and the way to conquer, theknowledge learnt from this course, further improvement suggestion.

2. **Length**:

The report should be at most 12 pages from cover to cover including diagrams/Testing results/references/appendix, if there is any. If you could wellpresent your work in fewer than 12 pages, you are encouraged to do so.

DO NOT include source code in the report but stored the source code in a folder. You are to ensure that the diagrams are readable and clear to the reader. [You can save the diagrams as image files and include in a folder]

APPENDIX D:

Creating Javadoc:

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

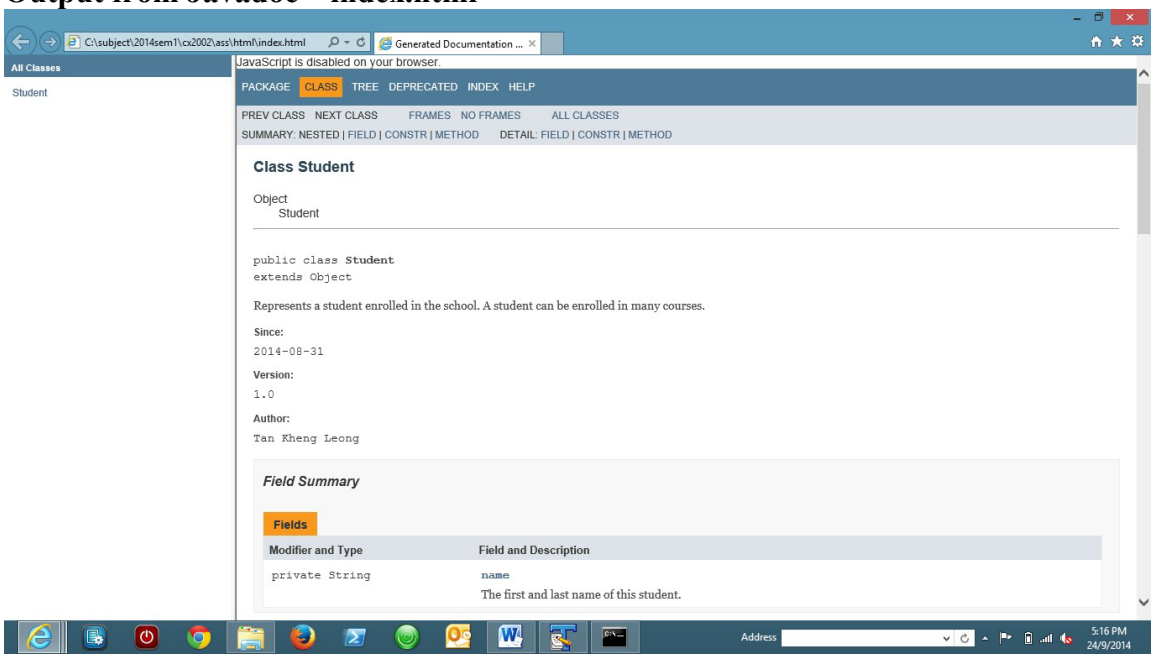
Using Javadoc in Eclipse : Youtube : http://www.youtube.com/watch?v=Hx-8BD_Osdw

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 folderto 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