



ASSIGNMENT

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

Building an OO Application

**SCHOOL OF COMPUTER SCIENCE AND ENGINEERING
NANYANG TECHNOLOGICAL UNIVERSITY**

1. **OBJECTIVE**

The main objective of this assignment is to:

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

2. **LABORATORY**

Free access only when necessary.

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 **Console-based application (non-Graphical UI)**:

MOvie Booking and LIsting Management Application (MOBLIMA)

MOBLIMA is an application to computerize the processes of making online booking and purchase of movie tickets, listing of movies and sale reporting. It will be used by the movie-goers and cinema staff.

The following are information about the application:

- The application acts as a centralized 'location' for making bookings for all the Cineplexes in different locations managed by the vendor.
- Each Cineplex will have 3 or more cinemas.
- The movie ticket price can be charged according to the following type :
 - a. type of movie (2D, Blockbuster, etc.),
 - b. class of cinema (e.g. Platinum Movie Suites)
 - c. age of movie-goer (e.g. adult, senior citizen, child)
 - d. day of the week or public holiday

An example is shown in Appendix A.

- Movie listings and showtimes can be queried and booking can be made.
- From the movie listings, movie-goer can also view the information about the movie. Information like

- a. Movie title
- b. Showing status [Coming Soon, Preview, Now Showing]
- c. SYNOPSIS – movie abstract
- d. Director
- e. Cast (at least 2)
- f. Overall reviewer rating (1 – 5 [best])
- g. Past reviews and reviewers' ratings

Movie-goer can also enter his/her review and rating.

[The overall rating is the *average (to 1 decimal place)* of the all individual ratings for the particular movie. **Overall reviewer rating will only be displayed if there are more than ONE individual rating, else “NA” is displayed**]

- A layout of each cinema will be presented for seat selections upon booking.
[the layout will contain aisle and main stairway. Refer to Appendix A for reference]
- Upon booking, the application will capture the movie-goer's name, mobile number and email address. Each payment will have a transaction id (TID). The TID is of the format XXXYYYYMMDDhhmm (Y : year, M : month, D : day, h : hour, m : minutes, XXX : cinema code in letters). [you will decide on the cinema code]
- Movie-goer can check their history of bookings.
- For **cinema staff** only :
 - a. Cinema staff needs to **login** to access its functions
 - b. Cinema staff can configure the system settings (e.g., ticket prices, holidays, etc.)
 - c. Cinema staff can enter the forthcoming movies, its type (Blockbuster/3D, etc.), movie rating (e.g. PG), show times, the cinema, showing status (Coming Soon, Preview, Now Showing, End Of Showing), etc.
 - d. Cinema staff can also update the details of the movies or remove the movie by changing the status to 'End of Showing'.
 - e. Cinema staff can list the current top 5 ranking movies by
 - i. Ticket sales (display the movie title and total sales)
 - ii. Overall reviewers' rating (display the movie title and overall rating)

Refer to Appendix A for further details and additional information can be acquired using the site, <http://www.cathaycinplexes.com.sg/default.aspx>, as reference.

Functional Requirements (to be demonstrated):

Admin module

1. Login
2. Create/Update/Remove movie listing
3. Create/Update/Remove cinema showtimes and the movies to be shown
4. Configure system settings

Movie-goer module

1. Search/List movie
2. View movie details – including reviews and ratings
3. Check seat availability and selection of seat/s.
4. Book and purchase ticket
5. View booking history
6. List the Top 5 ranking by ticket sales OR by overall reviewers' ratings

(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 File/I/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 can populate your movies 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

- configuring a holiday date and the ticket price is shown correctly when booking is done on that date
- booking on a different day of the week or holiday and type of cinema (eg suite to demonstrate the differences in prices
- configuring “End of Showing” date and the movie should not be listed for booking
- booking only allowed for “Preview” and “Now Showing” status.

***The test cases above need to be demonstrated.**

Assumptions :

- (1) This is a single-user application and there is no need to consider concurrent access.
- (2) THREE cineplexes will be created for the demonstration.
- (3) The currency will be in Singapore Dollar (SGD) and inclusive of Good and Services Tax (GST).
- (4) A simple login for cinema staff is sufficed.
- (5) Payment will always be successful.
- (6) There is no need to interface with external system, e.g. Payment, printer, etc., but you can consider it in your design.
- (7) Senior citizen can be purchased online without validation of identity or age. The validation will be done upon entering the cinema.

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) Screen captures of the testings done (those essential test cases not covered in your demo).
- c) A write-up on your design considerations, principles and the 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)
- a) [**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. The group must also email the WBS.xls to the course-coordinator with ALL members cc 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. Please introduce your members and group number at the start of video, 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 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 be done in real-time and NOT pre-run display.**

7. THE DELIVERABLE

Your group submission should include the following:

- a) The report (separate diagram file if diagram is unclear in report)
- b) Video and audio recording of the demonstration.
- c) All implementation codes and java documentation (javadoc).
- d) Other relevant files (eg data files, setup instruction, etc)

8. ASSESSMENT WEIGHTAGE

UML Class Diagram [30 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.

Design Consideration [15 Marks]

- Usage of OO concepts and principle - correctness and appropriateness

Implementation Code [35 Marks]

- Diagram to Code correctness, readability, Jav naming 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

9. DEADLINE

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 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, mpeg, wmv or mp4]

[In the event your video file is too large to upload, you may upload to **Youtube** and **provide the link in your report**. **No online storage** (like DropBox, Google Drive, etc) is allowed – no updates after submission].

2. **DEADLINE** : 13th November 2022 (Week 13 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/>
- NTULearn SC2002 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
- Windows Media Encoder (a suggestion)
http://www.microsoft.com/expression/products/EncoderPro_Overview.aspx

APPENDIX A:

Ticket Prices

01. Current ticket prices		
Non-Blockbuster Rates (2D)		
TICKET TYPE	Regular & Digital Movies	3D Movies
Senior Citizens*	\$4.00	N.A
Mon - Fri, before 6pm	Free Tea / Coffee	
Students**	\$7.00	\$9.00
Mon - Fri, before 6pm	Free 12oz Coke	
Mon - Wed#		
All sessions	\$8.50	\$11.00
Thu		
All sessions	\$9.50	\$11.00
Fri		
Sessions before 6pm	\$9.50	\$15.00
Fri		
Sessions from 6pm	\$11.00	\$15.00
Sat & Sun^		
All Sessions	\$11.00	\$15.00
Preferred Credit & Loyalty Cards^^	\$7 - \$9.50	N.A

**Not valid on PH/eve of PH ^Include PH/eve of PH and weekend sneaks *For patrons 55 years & older. Not valid on PH/eve of PH
^^Refer to cathay.com.sg (under FAQs) for list of preferred cards #Sneak preview at \$9.50 between Mon to Wed, excluding PH/eve of PH

Tickets for movies denoted as "Blockbuster" will be charged at \$1 more than the prevailing rate.

Screen 0

J
H
G
F
E
D
C
B
A

J
H
G
F
E
D
C
B
A

Entrance

LEGEND

 Seat Selected	 Seat Available
X Seat Taken	
 Couple Seat Selected	 Couple Seat Available
XX Couple Seat Taken	
 Elite Seat Selected	 Elite Seat Available
XXX Elite Seat Taken	
 Ultima Seat Selected	 Ultima Seat Available
XXXX Ultima Seat Taken	

Please note that this movie is rated **M18**.

Click [here](#) for more information about movie ratings.

Cathay Cineplexes welcomes all guests aged 18 and above to the Platinum Movie Suites.

CONFIRM SEATS
BACK TO MOVIE SELECTION

Please note that the system will not allow you to leave a single unoccupied seat between selected seats. ?

APPENDIX B:

Declaration of Original Work for SC/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
- Testing.
 - Test Cases and Results

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 well present 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;

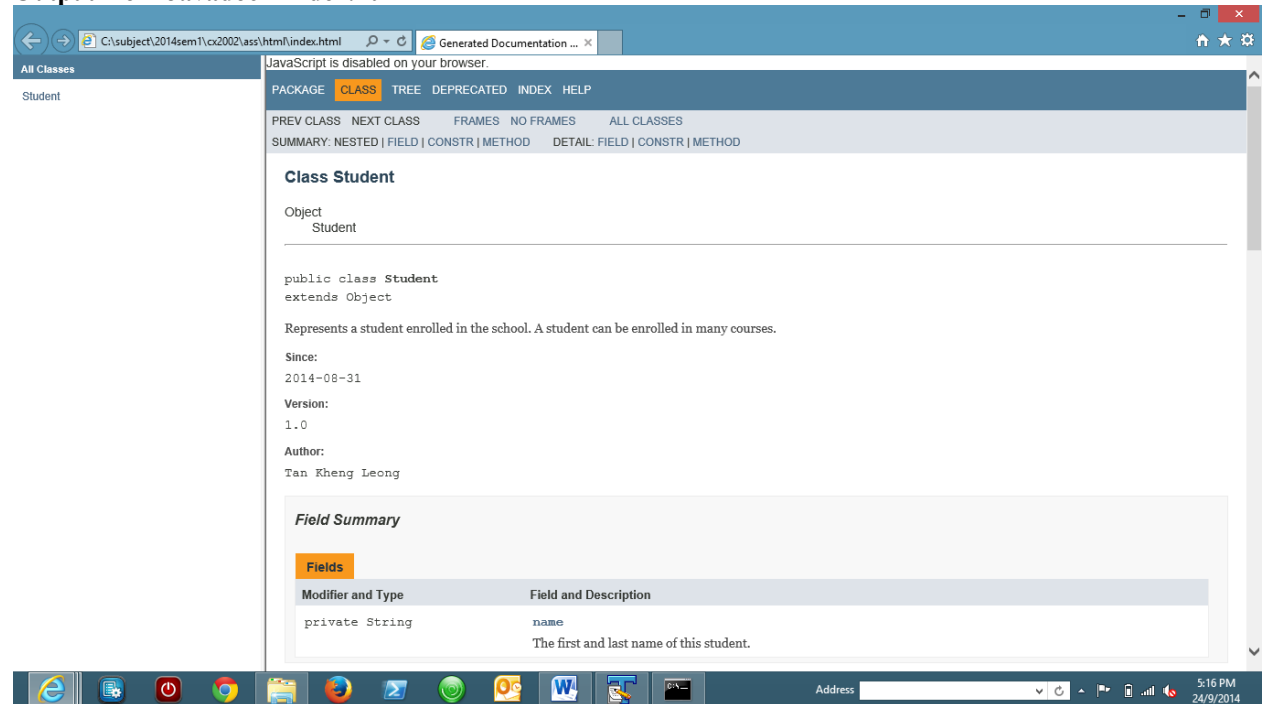
    /**
     * The age of this student.
     */
    private int age;

    /**
     * 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



For those familiar with using command prompt :

Steps to generate 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