# UCCD2044 Assignment

# Term: June 2023

# 1. General Instructions

The general guideline for this assignment is as follows:

1. Evidence of **plagiarism or collusion** will be taken seriously, and University regulations will be applied fully to such cases, in addition to **ZERO marks** being awarded to all parties involved.
2. The total marks for the assignment is 100 and contributes 20% to the total grade.
3. This is a **group assignment of 3 persons**.
4. The deadline for the submission of reports is on **Sept 8 (Friday) before 11.59 pm (Week 12)**.
5. Each group must submit ONE (1) **complete source code (src) with UML diagram (word or pdf)** and ONE (1) **recording video (10 mins max)** to explain each classes function and the output of your application. Please **compress all the files** and rename with your name, student id and program. Marks will be allocated based on your contribution. (Refer to section 3.4)

**Example:** *GroupXX\_CS\_Assignment*

1. Your file must be submitted to the following platform before the **due time/date**.

Send your answer script to Email Account:

* + 1. CN students please send your assignment to:  
       [**UCCD2044FACN@gmail.com**](about:blank)
    2. CS students please send your assignment to:  
       [**UCCD2044FACS@gmail.com**](about:blank)
    3. CT students please send your assignment to:  
       [**UCCD2044FACT@gmail.com**](about:blank)
    4. IA students please send your assignment to:  
       [**UCCD2044FAIA@gmail.com**](about:blank)
    5. IB students please send your assignment to:  
       [**UCCD2044FAIB@gmail.com**](about:blank)

**Note:** Use your “1UTAR” email account to submit your assignment. For the title of your email, please use the file name of your compress file name\_ Assignment. That is,

*GroupXX\_CS\_Assignment*

1. **Late submission** after the due time/date may incur a late penalty as shown below:

i. 0 hour < lateness ≤ 0.5 hour: 50%-mark deduction

ii. 0.5 hour < lateness ≤ 1 hours: 100%-mark deduction

# 2. Background

In this assignment, you are required to develop a Hospital Management System (HMS) in Java using Eclipse IDE and object-oriented programming concepts. The main objective of the HMS is to manage activities in a hospital, including patient, doctor, administrative staff, medicine, facility, and laboratory. And thus, this HMS should contain tables for all records of the patient, doctor, administrative staff, medicine, facility, and laboratory.

# 3. Requirements

Your assignment is to develop a program to manage the HMS described in the previous section.

**3.1 Program Functions (6% - 30 marks)**

Your program must fulfill the following functional specifications:

* Display the Welcome to the HMS and the current date and time.
* Display the main menu contains all the six selections (Doctors, Patients, Medical, Laboratories, Facilities, Staff).
* Declare, create and initialize array/array List (dynamic array) named doctors, patients, medicals, laboratories, facilities, and staffs. You are free to give any name during array initialization, but you are required to initialize at least **five values** for each of the array. **First id** and **names** in the doctors, patients, staffs should be **your name** and **id (last three number of your student id)** . For example, for the doctors array, the expected output as follows:

----------------------------------------------------------------------

id Name Specialist Timing Qualification Room No.

----------------------------------------------------------------------

001 Dr.Tom Wong Surgeon 8-11AM MBBS,MD 11

002 Dr.John Lim Physician 10-3AM MBBS,MS 45

003 Dr.Amy Chia Surgeon 7-11AM MBBS,MD 8

8

* Provide intuitive input method that allows users to add new entry or draw one or existing list from corresponding tables of the classes.
* Check if user would like to return to previous section or return to main menu.

**3.2 Object-Oriented Design (6% - 30 marks)**

You much use Java and object-oriented programming techniques in your implementation. Your design should follow good object-oriented design principles such as:

* Single responsibility – a class should have responsibility for a single functionality of a program and that responsibility should be encapsulated by the class.
* Open/closed principle – classes or modules should be open for extension but closed for modification.
* Efficient and no redundancy – keep it simple.

The followings are the basic classes that **MUST** exist in your program. The responsibilities of each of the classes, as well as some of their properties and methods are suggested. You are free to select the data types and method signatures for the classes, and to add additional classes if necessary.

## Staff:

**Staff** is the class for representing a staff object. The Staff class contains:

* Five String data fields named id, name, designation and sex. A field named salary that stores salary (int) of the staff.
* A method named newStaff()that prompts user to enter staff’s Id, name, designation, sex and salary.
* A method name showStaffInfo() to show the content of the staff as:

[id] [name] [designation] [sex] [salary]

## Doctor:

**Doctor** is the class for representing a doctor object. The **Doctor** class contains:

* Five String data fields named id, name, specialist, workTime, and qualification. A field named room that stores room number (int) of the doctor.
* A method named newDoctor()that prompts user to enter doctor’s Id, name, specialization, work time and qualification.
* A method name showDoctorInfo() to show the content of the doctor as:

[id] [name] [specialist] [work time] [qualification]

A screenshot of a computer program

Description automatically generated

## Patient:

**Patient** is the class for representing a patient object. The **Patient** class contains:

* Five String data fields named id, name, disease, sex, and admitStatus. A field named age that stores age (int) of the patient.
* A method named newPatient()that prompts user to enter patient’s Id, name, disease, sex, admit status and age.
* A method name showPatientInfo() to show the content of the patient as:

[id] [name] [disease] [sex] [admitStatus]

A screenshot of a blue screen

Description automatically generated

## Medical:

**Medical** is the class for representing a medical object. The **Medical** class contains:

* Three String data fields named name, manufacturer, and expiryDate. Two int data fields named cost and count.
* A method named newMedical()that prompts user to enter name, manufacturer, expiry date, cost and number of unit.
* A method name findMedical() to show the content of the medical as:

[name] [manufacturer] [expiryDate] [cost]

A blue screen with white text

Description automatically generated

## Lab:

**Lab** is the class for representing a lab object. The **Lab** class contains:

* A String data fields named lab. A data fields named cost that stores cost (int) of the facility.
* A method named newLab()that prompts user to enter facility and cost.
* A method name labList() to show the content of the lab as:

[lab] [cost]

A screenshot of a computer menu

Description automatically generated

## Facility:

**Facility** is the class for representing a facility object. The **Facility** class contains:

* A String data fields named facility.
* A method named newFacility()that prompts user to enter facility.
* A method name showFacility() to show the content of the facility as:

[facility]

A screenshot of a computer

Description automatically generated

## HospitalManagement:

The **HospitalManagement** class is the application class which controls the flow of the HMS, displays messages and gets inputs from the users. It should at least have the following attributes and methods:

* An array of **Doctor** objects. Number of doctors in the hospital is 25.
* An array of **Patient** objects that can hold 100 values.
* An array of **Lab** objects that can hold 20 values.
* An array of **Facility** objects that can hold 20 values.
* An array of **Medical** objects that can hold 100 values.
* An array of **Staff** objects that can hold 100 values.
* Provide existing lists for doctors, patients, labs, facilities, medicals and staffs, with a minimum number of three objects for each class.

**3.3 Interface (2% - 10 marks)**

* GUI enhancement

**3.4 Recording Video – Each group members MUST be in the presentation (6% - 30 marks)**

* Introductions: Your name, student IDs, title of assignment and show face.
* Program description, including what is the program about and how it works.
* UML Class diagrams showing the program structure.
* List the effort and contribution of each team member in an example table below:

|  |  |  |
| --- | --- | --- |
| Student 1 | Student 2 | Student 3 |
| Staff, Doc & Patient | Medical, Labs & Facility | Hotel management GUI |