# Group Assignment 3- Office Management System

For this assignment you are free to change your group and associate with a new student if you wish. However, I do not recommend that.

For this assignment, you are required to develop an office management system for a dentist office that only accepts cash with the following minimum requirements:

## Functional Requirements

1. It should display all the possible services offered to patients (fee schedule) in this specific office and how much patients should be charged for each service.
2. It should display a list of current patients.
3. It should display a list of doctors along with their specialization.
4. It should display a list of doctor assistants along with doctors they are assigned to.
5. It should enable the user to modify the fee schedule if necessary (add new services, delete and or modify current services).
6. It should enable the user to modify the patient list if necessary (add new patients, delete and or modify current ones).
7. It should enable the user to modify the doctor list if necessary (add new doctors, delete and or modify current ones).
8. It should enable the user to modify the doctor assistant list if necessary (add new assistants, delete and or modify current ones).
9. It should follow a logical business process.
10. It should store and display user modifications as long as the program is running (not terminated)
11. Your program should be able to handle any exception that may occur during the course of user interaction without crashing or getting stuck in an infinite loop.
12. Create an invoicing mechanism where the system asks for which patient you want to create an invoice, then ask about services that are performed on the patient and then generate an invoice with the patient information on top, list of services performed in the middle and a total due amount on the bottom
13. Unless the user specifically wants to end the program, the program should not end on its own and has to stay running all the time until user instruct it to end/exit.

## Non-functional Requirements:

1. You have to demonstrate your understanding of inheritance in Object Oriented programming through this project. At least one hierarchical class structure must be present in your application.
2. your code needs to be well documented and readable (comment extensively)
3. There are no set requirements for the UI. It can be either text based or graphical.

You need to make sure that your program meets the minimum requirements listed above. You can find an illustration of an acceptable final product at the following link: <https://youtu.be/R0MtAo7wssQ>

# Grading Criteria:

You will receive a grade if only you can demonstrate that your program runs without any errors as it should. You will receive no credit (**Zero**) if your program fails to run and work as expected.

* Look and feel: 15 pts
* Meeting functional and non-functional requirements (found on instructions): 85pts
  + You can lose up to 15 points for each requirement that is not met.
* Only one member of the group needs to submit the required files
* Clearly state who the other group member is through the submission.
* Failure to comply with the above instructions will result in minimum of 5 points deduction.

# Submission Instruction

You can create either an android application or a java program.

## Java Program Submission Instructions:

* Submit only exported archive file of your project
* To export an archive file on Eclipse consider the following process
  + Select the project you want to export
  + Go to the “File” menu and select export.
  + In the export dialog, select General and then Archive File
  + Hit next, make sure your project is selected correctly, give it a path and hit finish
  + Locate the archive file and submit it.

## Android Submission Instructions:

You must rename your .java and .xml files to .txt and submit the following items to the D2L dropbox before the deadline:

1. Your main activity java file and other Java classes in your application
2. Your main layout xml file and all of related layout xml files
3. Your entire resource folder files (e.g. strings.xml, colors.xml, etc.)
4. Your signed application .apk file (release version)

# Tips for coding

* ArrayLists are your best friend in developing this program in a Java environment
* You can also utilize databases, though I wouldn’t recommend it.
* If you decide to develop this system in an android environment, you can utilize Android SQLite.