

**Assignment #1**  
COMP 303  
Well-Formed Networks of Objects  
Due: October 4<sup>th</sup>, 2016

All your solutions, whether programmed or designed on paper, must follow three important programming rules, they must be: well-formed, optimal, and energy conservative. These three rules are defined below:

Well-formed – Each program and diagram must follow proper syntax and adhere to the strict object oriented principals of encapsulation, information hiding and interfacing (API's that protect).

Optimal – Each solution must be efficient in execution speed and employ an economy of memory.

Energy conservative – Each solution must have a simple design, keep its lines of code to a minimum (or minimum arcs/boxes, or text), and require the least effort to code (or draw, or write). BUT, at the same time, maximize its expressiveness (i.e. being clear in its meaning).

For this assignment you will not submit compiled code. Submit your assignment as a PDF document containing answers to the following questions:

Question 1: Well-Formed Object (25 points)

To answer this question, create an executable Java program having two classes: Student and Queue. Both classes must be well-formed.

A high school graduation party only permits valid high school students over the age of 15. They ask you to build a Java program that has two important classes: Student and Queue. The Student class contains the student's name, age and school ID card. A valid student is 15 or over and their 5-digit ID begins with the number 22, as in 22765 and 22156. The Queue class is a standard queue implementation that provides the following functionality: enqueues, dequeues, and stores only Student objects. The graduation party organizers do not know how many people will come to the party. The program does not use files. The purpose of the queue is to manage the lineup of people as they wait to enter the gym, where the party will be held.

For this question do the following:

- Submit two well-formed Java files (Student and Queue)

Question 2: Well-Formed Use-Case with Template (25 points)

ABC Airlines would like to manage their planes and flight routes. They have 4 airplanes and they fly to 10 destinations. Planes are either in-flight or at a location (source or destination). Each plane has statistics they would like to track: size of flight crew, maximum passenger capacity, number of passengers, the route, and where they are in that route. A route has two locations: a beginning location and an end location. They would like to be able to create and delete airplanes, create and delete routes, assign a route to an airplane, edit the airplane statistics. The airline is required by law to have 20% more employees on staff than are needed to manage the airplanes they possess. This application tracks the employees (name, rank and position) and to which airplane they are currently

assigned to. This application also tracks passengers (name and boarding pass). Only passengers possessing a valid boarding pass can be on an airplane. They would like to be able to create and delete people, assign or remove them to/from an airplane, and set the validity of the passenger's boarding pass.

For this question do the following:

- Provide a solution use-case diagram
- Provide a solution use-case template for the most complex activity

Question 3: Design an optimal Well-Formed Network-of-objects using a Domain Model (25 points)

Company ABC wants their 20 employees to punch in and out from work using a computer. ABC builds between 10 to 100 widgets per day from raw material. An employee punches in when they pick up raw material to build a single widget. They also input their employee ID. The system validates their ID before accepting the punch-in. After they finish building a widget, they place the finished widget in the shipping room and punch out. If they plan to build another widget from raw material, they then get additional raw material and they punch in again. If they punched in and constructed more than one widget there is an optional place to enter the number of widgets created, otherwise it assumes one widget. At the end of the day the manager, using another computer over a local area network, generates a report of the number of widgets created for a period of time by employee. The manager must also enter their ID number, which is also validated by the system, before they can use it. By default, the system assumes the current date but optionally the user can specify a date range. The report lists each employee and the quantity of widgets they built within the specified period.

There is more information in this question than you need in your answer, so choose wisely.

For this question do the following:

- Provide a solution domain model

Question 4: Optimal State-Transition or Interaction Diagram (25 points)

Using the problem statement from question 3, decide which diagram is better to use (state diagram or interaction diagram) to describe how the punching in-out system functions. Then, using the diagram you selected, describe the problem statement's punching in-out system. 50% of the points are given to selecting the better diagram to use and the remaining 50% goes to the correctness of the diagram.

For this question do the following:

- Select the best diagram to use
- Using your selected diagram, describe the punch in-out system