ESOF 322: Software Engineering I **DUE Date: October 2, 2018**

Instructions (45 pts):

- Do all exercises with your partner.
- Clearly print the names of all participants in the first page of your assignment.
- No hand-written answers allowed.
- Absolutely no late assignments.
- Your assignment should be turned in to D2L

Exercise 1 (10 pts)

Calculate the Function Point value for the following requirements. Show **all** of your work.

Train Station

You are required to build a software system to handle the dispensing of tickets at a train station. In this case the ticket-dispensing machine can hold only a finite number of coins and bills. The machine stores only information about trains that leave the station in which the machine is located. Each train makes a finite number of stops after leaving the station. These stops are called destinations. Each train has two types of cabins: first class and regular class. The maximum number of seats in each cabin varies across trains. A user of the ticket machine must enter a destination (as a selection), number of tickets required, and the type of cabin requested (only one type of cabin can be selected for the number of tickets ordered). The machine then checks whether a train that stops at the destination has available space in the type of cabin requested. If there is an available train, the system then reserves the required number of seats and displays the price to the user. The user must then enter payment (restrictions on type of payment are given below). After sufficient payment is entered the system dispenses the tickets. Each ticket contains the train identifier, the cabin type, and a seat number. If insufficient payment is entered the reservation is cancelled. The machine operates under the following conditions:

- This machine accepts only cash. Only nickels, dimes, quarters, dollar bills and five-dollar bills are to be accepted as valid contributions to a payment.
- All other objects are rejected (rejected objects are called slugs).
- The number of coins and bills that the machine can store is limited. Below are the limits:
  - Nickels: maxN
  - Dimes: maxD
  - Quarters: maxQ
  - Dollar bills (1 and 5 dollar bills combined): maxB
- A ticket can only be dispensed if it is available and the payment is sufficient.

Payment can only be made after a reservation is made on an available train

Exercise 2 (10 pts)

We have covered three design patterns in class. Pick two design patterns (any two that you want –you can even do research on your own and find another one you like!) and couple them.

Two design patterns are coupled when at least one class is a participant in both design patterns.

a) Draw a UML class diagram that clearly shows the coupling. Please identify which class(es) participates in multiple patterns.
b) Draw a UML Sequence diagram where you demonstrate the behavior of an instance of the coupled class from the perspective of one pattern, then from the other pattern.

Exercise 3 (5 pts)

Explain the "extend" and "include" relationships in UML Use Case diagrams. Provide a Use Case diagram with an example that uses both relationships.

Exercise 4 (10 pts)

We have discussed a number of different types of UML class diagram relationships in class.

a) List all the relationships
b) Order your list such that the relationships are listed according to their refactoring complexity. In other words, start with the relationship that would be easiest to refactor and finish with the relationship that would be the hardest to refactor. Explain why you chose this order.

Exercise 5 (10 pts)

1)

a) Draw a UML class diagram of a binary tree. Each node contains an integer.
b) Provide the corresponding object oriented code that implements your binary tree design.

2)

a) Draw a UML class diagram of a linked list that contains Employee records as data. An Employee record has a name, a social security number, and a salary.
b) Provide the corresponding object oriented code that implements your linked list.

Note: Only provide the code of the structures. Do not write code to exercise your data structures.

Hand in:

- Bring a printout to class that has answers to all 5 exercises.
- Hand in your answers (as a single file) in D2L