ESOF 322: Software Engineering I **DUE Date: September 26, 2019**

Instructions (24 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)

We have covered three design patterns in class (Strategy, Adapter, Observer). 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 perspective of the other pattern.

Exercise 2 (14 pts)

Short answer (2 points each):

a) Assume during your team's last sprint, that they completed 32 story points using a 3-person team working in sprints of 3 weeks for a total of 45-man days. Calculate your team's estimated velocity for the next sprint if we still have 3-week sprints, but you now added 2 engineers to the team, and one of them can only work 80% of the time.
b) How would you estimate a focus factor for a brand-new team?
c) We looked at using poker using semi-Fibonacci sequences to estimate story points. Think of another way to estimate story points and explain it. Is it better or worse than poker?
d) Draw a UML class diagram of a binary tree. Each node contains an integer.
e) Provide the corresponding object-oriented code that implements your binary tree design from part d.
f) 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.
g) Provide the corresponding object-oriented code that implements your linked list from part f.

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 exercises.
- Hand in your answers (as a single file) in D2L