Name: Casper Kristiansson **Student#:** 20938643 **Lec. sec.:** L17

Date: 2022-11-16

COMP 3111: Software Engineering

Lecture 17 Exercises: Design Patterns

Exercise 1: Which of the following are disadvantages of using inheritance to provide Duck behaviour (Choose all the apply.)?

- ☐ A. Code is duplicated across subclasses.
- ☒ B. Runtime behaviour changes are difficult.
- ☐ C. We can't make ducks dance.
- ☒ D. Hard to gain knowledge of all duck behaviours.
- ☐ E. Ducks can't fly and quack at the same time.
- ☒ F. Changes can unintentionally affect other ducks.

Exercise 2: How does the observer pattern use the following principles?

Design Principle: Identify the aspects of your application that vary and separate them from what stays the same.

Encapsulate different classes so that the actual instances of a specific class need to change all of the instances won't change.

Design Principle: Program to an interface, not an implementation.

That way the duck classes won't need to know any of the implementations of others. Meaning that you shouldn't hardcode which makes it more difficult to change during runtime. Meaning that you want to make it more flexible. For example, not hardcoding a class for a dog but rather have it inherited from an animal class.

Design Principle: Favour composition over inheritance.

By using composition over inheritance because the design becomes a lot more flexible which in practice will make the code more reusable.